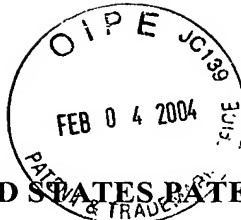


Docket No. 242756US2/ims



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Kunihiro AKIYOSHI, et al.

GAU: 2853

SERIAL NO: 10/660,723

EXAMINER:

FILED: September 12, 2003

FOR: IMAGE FORMING APPARATUS AND METHODS USED IN THE IMAGE FORMING APPARATUS

REQUEST FOR PRIORITY

COMMISSIONER FOR PATENTS
ALEXANDRIA, VIRGINIA 22313

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number , filed , is claimed pursuant to the provisions of 35 U.S.C. §120.
- ☐ Full benefit of the filing date(s) of U.S. Provisional Application(s) is claimed pursuant to the provisions of 35 U.S.C. §119(e): Application No. Date Filed
- ☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>MONTH/DAY/YEAR</u>
JAPAN	2002-269280	September 13, 2002
JAPAN	2002-269281	September 13, 2002
JAPAN	2002-269282	September 13, 2002
JAPAN	2002-307636	September 14, 2002
JAPAN	2002-307637	September 14, 2002
JAPAN	2003-317280	September 9, 2003
JAPAN	2003-317281	September 9, 2003
JAPAN	2003-317282	September 9, 2003
JAPAN	2003-317283	September 9, 2003

Certified copies of the corresponding Convention Application(s)

- ☒ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. filed
- ☐ were submitted to the International Bureau in PCT Application Number
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. filed ; and
- ☐ (B) Application Serial No.(s)
- ☐ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee


Customer Number

22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 05/03)

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.


Marvin J. Spivak

Registration No. 24,913

Joseph A. Scafetta, Jr.
Registration No. 26, 803

10/660, 723

日 本 国 特 許 庁
JAPAN PATENT OFFICE

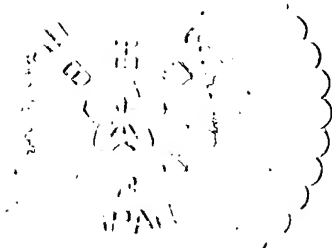
別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 2 年 9 月 1 3 日
Date of Application:

出 願 番 号 特 願 2 0 0 2 - 2 6 9 2 8 0
Application Number:
[ST. 10/C]: [J P 2 0 0 2 - 2 6 9 2 8 0]

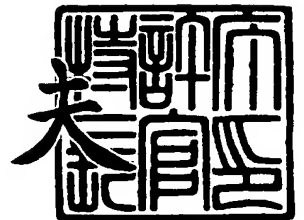
出 願 人 株 式 会 社 リ コ ー
Applicant(s):



2 0 0 3 年 8 月 1 1 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 3 - 3 0 6 4 2 1 3

【書類名】 特許願

【整理番号】 0203936

【提出日】 平成14年 9月13日

【あて先】 特許庁長官殿

【国際特許分類】 G03G 21/00 370

【発明の名称】 画像形成装置およびアプリ起動制限方法

【請求項の数】 28

【発明者】

 【住所又は居所】 東京都大田区中馬込 1 丁目 3 番 6 号 株式会社リコー内

 【氏名】 秋吉 邦洋

【発明者】

 【住所又は居所】 東京都大田区中馬込 1 丁目 3 番 6 号 株式会社リコー内

 【氏名】 田中 浩行

【特許出願人】

 【識別番号】 000006747

 【氏名又は名称】 株式会社リコー

【代理人】

 【識別番号】 100089118

 【弁理士】

 【氏名又は名称】 酒井 宏明

【手数料の表示】

 【予納台帳番号】 036711

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9808514

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 画像形成装置およびアプリ起動制限方法

【特許請求の範囲】

【請求項 1】 画像形成処理にかかるユーザサービスにそれぞれ固有の処理を行うアプリケーションと、前記アプリケーションと画像形成処理で使用されるハードウェア資源との間に介在し、前記ハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うコントロールサービスとを備えた画像形成装置であって、

前記画像形成装置に特有のアプリケーション以外のアプリケーションを起動する前に、前記画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得し、取得したリソース量情報と、上記起動アプリケーションが使用するリソース使用量に関するリソース使用情報とに基づいて、前記リソース使用量が前記リソース量を越えている場合は前記起動アプリケーションの起動を制限するアプリ起動制限手段、

を備えたことを特徴とする画像形成装置。

【請求項 2】 前記画像形成装置に特有のアプリケーション以外の起動アプリケーションが使用するリソース使用量に関する情報を取得して前記リソース使用情報を生成するリソース使用情報取得手段をさらに備えたことを特徴とする請求項 1 に記載の画像形成装置。

【請求項 3】 前記リソース使用情報取得手段は、前記起動アプリケーションが自ら保有するリソース使用量に関する情報を取得して前記リソース使用情報を生成することを特徴とする請求項 2 に記載の画像形成装置。

【請求項 4】 前記リソース使用情報取得手段は、前記起動アプリケーションを仮起動することにより、当該起動アプリケーションが使用するリソース使用量に関する情報を取得して前記リソース使用情報を生成することを特徴とする請求項 2 に記載の画像形成装置。

【請求項 5】 前記リソース使用情報取得手段は、前記起動アプリケーションが自ら保有するリソース使用量、あるいは、前記起動アプリケーションの仮起動によるリソース使用量に基づいて通常起動を行い、その通常起動時に当該起動

アプリケーションが実際に使用するリソース使用量に基づいて前記リソース使用情報を生成することを特徴とする請求項 3 または 4 に記載の画像形成装置。

【請求項 6】 前記リソース使用情報は、前記起動アプリケーションの通常起動時に実際に使用されるリソース使用量の平均値としたことを特徴とする請求項 5 に記載の画像形成装置。

【請求項 7】 前記リソース使用情報取得手段は、前記起動アプリケーションが使用するメモリ領域情報を取得して前記リソース使用情報を生成し、

前記アプリ起動制限手段は、画像形成装置におけるメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するメモリ使用量を除いたメモリ領域残容量を前記リソース量情報として取得し、取得したメモリ領域残容量と、前記メモリ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする請求項 2 ～ 6 のいずれか一つに記載の画像形成装置。

【請求項 8】 前記リソース使用情報取得手段は、前記起動アプリケーションが使用するテキストメモリ領域情報を取得して前記リソース使用情報を生成し、

前記アプリ起動制限手段は、画像形成装置におけるテキストメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するテキストメモリ使用量を除いたテキストメモリ領域残容量を前記リソース量情報として取得し、取得したテキストメモリ領域残容量と、前記テキストメモリ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする請求項 7 に記載の画像形成装置。

【請求項 9】 前記リソース使用情報取得手段は、前記起動アプリケーションが使用するヒープ領域情報を取得して前記リソース使用情報を生成し、

前記アプリ起動制限手段は、画像形成装置におけるヒープ領域から画像形成装置特有のアプリケーションおよびシステムが使用するヒープ領域使用量を除いたヒープ領域残容量を前記リソース量情報として取得し、取得したヒープ領域残容量と、前記ヒープ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする請求項 7 または 8 に記載の画像形成装置。

【請求項 10】 前記リソース使用情報取得手段は、前記起動アプリケーシ

ョンが使用するスタック領域情報を取得して前記リソース使用情報を生成し、

前記アプリ起動制限手段は、画像形成装置におけるスタック領域から画像形成装置特有のアプリケーションおよびシステムが使用するスタック領域使用量を除いたスタック領域残容量を前記リソース量情報として取得し、取得したスタック領域残容量と、前記スタック領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする請求項 7～9 のいずれか一つに記載の画像形成装置。

【請求項 11】 前記リソース使用情報取得手段は、プロセスが使用するリソースに関する情報を保持するシステム情報から前記起動アプリケーションが使用するリソースに関する情報を取得し、前記リソース使用情報を生成することを特徴とする請求項 2～10 のいずれか一つに記載の画像形成装置。

【請求項 12】 前記アプリ起動制限手段は、前記リソース量情報と前記リソース使用情報に基づいて、起動前のアプリケーションに対して起動を制限する起動制限要求メッセージを送信することを特徴とする請求項 1～11 のいずれか一つに記載の画像形成装置。

【請求項 13】 前記コントロールサービスをサーバとしたクライアントプロセスとして動作し、かつ前記アプリケーションをクライアントとしたサーバプロセスとして動作する仮想アプリケーションサービスをさらに備え、

前記リソース使用情報取得手段と前記アプリ起動制限手段は、前記仮想アプリケーションサービスに含まれることを特徴とする請求項 2～12 のいずれか一つに記載の画像形成装置。

【請求項 14】 前記アプリ起動制限手段と前記リソース使用情報取得手段は、前記仮想アプリケーションサービスのプロセス内部でスレッドとして生成されることを特徴とする請求項 13 に記載の画像形成装置。

【請求項 15】 前記仮想アプリケーションサービスは、前記アプリケーション毎に設けられ、前記画像形成装置に特有のアプリケーション以外のアプリケーションを起動する前に、対応する仮想アプリケーションサービスのリソース使用情報取得手段とアプリ起動制限手段とを使って、画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報

と、起動するアプリケーションのリソース使用量に関するリソース使用情報とを取得し、リソース使用量がリソース量を越えている場合にアプリケーションの起動を制限することを特徴とする請求項 13 または 14 に記載の画像形成装置。

【請求項 16】 前記コントロールサービスは、前記ユーザサービスを提供する際に、アプリケーションの少なくとも 2 つが共通的に必要とする前記ハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うことを特徴とする請求項 1～15 のいずれか一つに記載の画像形成装置。

【請求項 17】 画像形成処理にかかるユーザサービスにそれぞれ固有の処理を行うアプリケーションと、前記アプリケーションと画像形成処理で使用されるハードウェア資源との間に介在し、前記ハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うコントロールサービスとを備えた画像形成装置上で、前記アプリケーションの起動制限を行うアプリ起動制限方法であって、

前記画像形成装置に特有のアプリケーション以外のアプリケーションを起動する前に、前記画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得し、取得したリソース量情報と、上記起動アプリケーションが使用するリソース使用量に関するリソース使用情報とに基づいて、前記リソース使用量が前記リソース量を越えている場合は前記起動アプリケーションの起動を制限するアプリ起動制限ステップ、

を含むことを特徴とするアプリ起動制限方法。

【請求項 18】 前記画像形成装置に特有のアプリケーション以外の起動アプリケーションが使用するリソース使用量に関する情報を取得して前記リソース使用情報を生成するリソース使用情報取得ステップをさらに含むことを特徴とする請求項 17 に記載のアプリ起動制限方法。

【請求項 19】 前記リソース使用情報取得ステップは、前記起動アプリケーションが自ら保有するリソース使用量に関する情報を取得して前記リソース使用情報を生成することを特徴とする請求項 18 に記載のアプリ起動制限方法。

【請求項 20】 前記リソース使用情報取得ステップは、前記起動アプリケーションを仮起動することにより、当該起動アプリケーションが使用するリソース使用量に関する情報を取得して前記リソース使用情報を生成することを特徴と

する請求項 18 に記載のアプリ起動制限方法。

【請求項 21】 前記リソース使用情報取得ステップは、前記起動アプリケーションが自ら保有するリソース使用量、あるいは、前記起動アプリケーションの仮起動によるリソース使用量に基づいて通常起動を行い、その通常起動時に当該起動アプリケーションが実際に使用するリソース使用量に基づいて前記リソース使用情報を生成することを特徴とする請求項 19 または 20 に記載のアプリ起動制限方法。

【請求項 22】 前記リソース使用情報は、前記起動アプリケーションの通常起動時に実際に使用されるリソース量の平均値としたことを特徴とする請求項 21 に記載のアプリ起動制限方法。

【請求項 23】 前記リソース使用情報取得ステップは、前記起動アプリケーションが使用するメモリ領域情報を取得して前記リソース使用情報を生成し、
前記アプリ起動制限ステップは、画像形成装置におけるメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するメモリ使用量を除いたメモリ領域残容量を前記リソース量情報として取得し、取得したメモリ領域残容量と、前記メモリ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする請求項 18 ～ 22 のいずれか一つに記載のアプリ起動制限方法。

【請求項 24】 前記リソース使用情報取得ステップは、前記起動アプリケーションが使用するテキストメモリ領域情報を取得して前記リソース使用情報を生成し、

前記アプリ起動制限ステップは、画像形成装置におけるテキストメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するテキストメモリ使用量を除いたテキストメモリ領域残容量を前記リソース量情報として取得し、取得したテキストメモリ領域残容量と、前記テキストメモリ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする請求項 23 に記載のアプリ起動制限方法。

【請求項 25】 前記リソース使用情報取得ステップは、前記起動アプリケーションが使用するヒープ領域情報を取得して前記リソース使用情報を生成し、

前記アプリ起動制限ステップは、画像形成装置におけるヒープ領域から画像形成装置特有のアプリケーションおよびシステムが使用するヒープ領域使用量を除いたヒープ領域残容量を前記リソース量情報として取得し、取得したヒープ領域残容量と、前記ヒープ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする請求項 23 または 24 に記載のアプリ起動制限方法。

【請求項 26】 前記リソース使用情報取得ステップは、前記起動アプリケーションが使用するスタック領域情報を取得して前記リソース使用情報を生成し

前記アプリ起動制限ステップは、画像形成装置におけるスタック領域から画像形成装置特有のアプリケーションおよびシステムが使用するスタック領域使用量を除いたスタック領域残容量を前記リソース量情報として取得し、取得したスタック領域残容量と、前記スタック領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする請求項 23～25 のいずれか一つに記載のアプリ起動制限方法。

【請求項 27】 前記リソース使用情報取得ステップは、プロセスが使用するリソースに関する情報を保持するシステム情報から前記起動アプリケーションが使用するリソースに関する情報を取得し、前記リソース使用情報を生成することを特徴とする請求項 18～26 のいずれか一つに記載のアプリ起動制限方法。

【請求項 28】 前記アプリ起動制限ステップは、前記リソース量情報と前記リソース使用情報に基づいて、起動前のアプリケーションに対して起動を制限する起動制限要求メッセージを送信することを特徴とする請求項 17～27 のいずれか一つに記載のアプリ起動制限方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、コピー、プリンタ、スキャナおよびファクシミリなどの画像形成処理にかかるユーザサービスを提供するアプリケーションのリソース使用量に応じてアプリケーションの起動の制限を行う画像形成装置およびアプリ起動制限方

法に関するものである。

【0002】

【従来の技術】

近年では、プリンタ、コピー、ファクシミリ、スキャナなどの各装置の機能を1つの筐体内に収納した画像形成装置（以下、「複合機」という。）が知られている。この複合機は、1つの筐体内に表示部、印刷部および撮像部などを設けるとともに、プリンタ、コピーおよびファクシミリ装置にそれぞれ対応した3種類のソフトウェアを設け、これらのソフトウェアを切り替えることによって、当該装置をプリンタ、コピー、スキャナまたはファクシミリ装置として動作させるものである。

【0003】

このような従来の複合機では、メモリなどの限られたリソース（資源）の範囲内でプリンタ、コピー、ファクシミリ、スキャナなどの各機能単位ですべてのアプリケーションプログラムが起動される。言い換えれば、従来の複合機では、すべてのアプリケーションプログラムを起動できる程度のリソースが用意されており、アプリケーションプログラムが起動不可能になる状況が生じる場合は想定されていなかった。

【0004】

【発明が解決しようとする課題】

ところで、このような従来の複合機では、プリンタ、コピー、スキャナおよびファクシミリ装置に対応するソフトウェアをそれぞれ別個に設けているため、各ソフトウェアの開発に多大の時間を要する。このため、出願人は、表示部、印刷部および撮像部などの画像形成処理で使用されるハードウェア資源を有し、プリンタ、コピーまたはファクシミリなどの各ユーザサービスにそれぞれ固有の処理を行うアプリケーションを複数搭載し、これらのアプリケーションとハードウェア資源との間に介在して、ユーザサービスを提供する際に、アプリケーションの少なくとも2つが共通的に必要とするハードウェア資源の管理、実行制御並びに画像形成処理を行う各種コントロールサービスからなるプラットフォームを備えた画像形成装置（複合機）を発明した。

【0005】

このような新規な複合機では、アプリケーションの少なくとも2つが共通的に必要とするサービスを提供するコントロールサービスをアプリケーションと別個に設けた構成となっているため、アプリケーションのサイズが従来の複合機で動作するアプリケーションプログラムのサイズに比べて小さなものとなっており、アプリケーションの起動および終了が頻繁に行われる。

【0006】

このため、複合機に搭載されているメモリなどのリソースの使用状況は頻繁に変化し、すべてのアプリケーションを起動した状態とすることができない場合も生じてくる。このような状況でアプリケーションを起動しても、アプリケーションが不正に終了してしまい、複合機の動作が不安定になるという問題がある。

【0007】

また、かかる新規な複合機は、アプリケーションとコントロールサービスとを別個に設けているため、複合機の出荷後にユーザもしくは第三者であるサードベンダが新規なアプリケーションを開発して複合機に搭載可能な構成となっている。このため、かかる複合機では、出荷時に搭載されているコピー、プリンタ、スキャナ、ファクシミリといった画像形成装置特有のアプリケーション以外に、従来の複合機とは異なるユーザやサードベンダが開発した新規アプリケーションなども多数起動可能となっている。このように、コピー、プリンタ、ファクシミリ、スキャナなど複合機であらかじめ提供されている特有のアプリケーションの場合は、限りのあるリソースを意識した構造となっているが、第三者が開発する新規アプリケーションの場合には、開発するベンダによって必ずしもリソースを意識した構造となっていない場合も考えられる。そこで、限りのあるリソースの範囲内でリソース使用量のわからない新規アプリケーションを起動させた場合、複合機の動作が不安定となる可能性が高く、さらに、アプリケーションが使用するリソース量が複合機で利用可能な残りリソース量を越えた場合は、リソース不足により複合機の動作が停止するという従来の複合機では問題にならなかった新規な課題が生じてくる。

【0008】

この発明は上記に鑑みてなされたもので、画像形成装置特有のアプリケーションやシステムの動作に必要なリソース量情報と、画像形成装置特有のアプリケーション以外の起動アプリケーションが使用するリソース使用量とに基づいて、アプリケーションを起動制限することにより、画像形成装置特有のアプリケーションを常に安定した状態で確実に動作させることのできる画像形成装置およびアプリ起動制限方法を得ることを目的とする。

【0009】

【課題を解決するための手段】

上記目的を達成するため、請求項1にかかる発明は、画像形成処理にかかるユーザサービスにそれぞれ固有の処理を行うアプリケーションと、前記アプリケーションと画像形成処理で使用されるハードウェア資源との間に介在し、前記ハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うコントロールサービスとを備えた画像形成装置であって、前記画像形成装置に特有のアプリケーション以外のアプリケーションを起動する前に、前記画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得し、取得したリソース量情報と、上記起動アプリケーションが使用するリソース使用量に関するリソース使用情報とに基づいて、前記リソース使用量が前記リソース量を越えている場合は前記起動アプリケーションの起動を制限するアプリ起動制限手段を備えたことを特徴とする。

【0010】

この請求項1にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーションが起動される前に、アプリ起動制限手段によって、画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得し、取得したリソース量情報と、起動アプリケーションが使用するリソース使用量に関するリソース使用情報とに基づいて、リソース使用量がリソース量を越えている場合に起動アプリケーションの起動を制限することで、リソースの状況によって画像形成装置特有のアプリケーションの実行ができなくなることを未然に回避し、画像形成装置のシステムの安定性を向上させることができる。

【0011】

また、請求項2にかかる発明は、請求項1に記載の画像形成装置において、前記画像形成装置に特有のアプリケーション以外の起動アプリケーションが使用するリソース使用量に関する情報を取得して前記リソース使用情報を生成するリソース使用情報取得手段をさらに備えたことを特徴とする。

【0012】

この請求項2にかかる発明によれば、リソース使用情報取得手段によって、画像形成装置に特有のアプリケーション以外の起動アプリケーションが使用するリソース使用量に関する情報を取得してリソース使用情報を生成することで、起動アプリケーションごとに実際に必要とするリソースに関するリソース使用情報に基づいてアプリ起動制限手段によるアプリ起動を制限することができ、画像形成装置のシステムの安定性をより向上させることができる。

【0013】

また、請求項3にかかる発明は、請求項2に記載の画像形成装置において、前記リソース使用情報取得手段は、前記起動アプリケーションが自ら保有するリソース使用量に関する情報を取得して前記リソース使用情報を生成することを特徴とする。

【0014】

この請求項3にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーション（ユーザやサードベンダが開発した新規アプリケーションなど）を起動する前に、リソース使用情報取得手段によって、各アプリケーションが自ら保有するリソース使用量に関する情報を取得し、リソース使用情報を生成することで、起動アプリケーションが使用する正確なリソース使用量を事前に把握することが可能となり、制限すべきアプリケーションの起動か否かを判断することができる。

【0015】

また、請求項4にかかる発明は、請求項2に記載の画像形成装置において、前記リソース使用情報取得手段は、前記起動アプリケーションを仮起動することにより、当該起動アプリケーションが使用するリソース使用量に関する情報を取得

して前記リソース使用情報を生成することを特徴とする。

【0016】

この請求項4にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーション（ユーザやサードベンダが開発した新規アプリケーションなど）を通常起動する前に、リソース使用情報取得手段によって、起動前のアプリケーションを仮起動させることにより、そのアプリケーションが使用するリソース使用量に関する情報を取得することができ、ここからリソース使用情報を生成することで、起動アプリケーションが使用する正確なリソース使用量を事前に把握することが可能となり、制限すべきアプリケーションの起動か否かを判断することができる。

【0017】

また、請求項5にかかる発明は、請求項3または4に記載の画像形成装置において、前記リソース使用情報取得手段は、前記起動アプリケーションが自ら保有するリソース使用量、あるいは、前記起動アプリケーションの仮起動によるリソース使用量に基づいて通常起動を行い、その通常起動時に当該起動アプリケーションが実際に使用するリソース使用量に基づいて前記リソース使用情報を生成することを特徴とする。

【0018】

この請求項5にかかる発明によれば、リソース使用情報取得手段によって、起動アプリケーションが自ら保有するリソース使用量、あるいは、起動アプリケーションの仮起動によるリソース使用量に基づいて通常起動を行い、通常起動時に起動アプリケーションが実際に使用したリソース使用量に基づいてリソース使用情報を生成することで、起動アプリケーションが実際に使用するより正確なリソース使用量が把握できるようになり、当該アプリケーションが再度起動される前に、制限すべきアプリケーションの起動か否かを判断することができる。特に、単独でのアプリケーションの起動では問題にならない程度のリソース使用量であっても、複数のアプリケーションを同時または追加的に起動する場合は、それらの合計リソース使用量が問題となるため、各アプリケーションが使用するリソース使用量をできるだけ正確に把握することが必要となる。

【0019】

また、請求項6にかかる発明は、請求項5に記載の画像形成装置において、前記リソース使用情報は、前記起動アプリケーションの通常起動時に実際に使用されるリソース使用量の平均値としたことを特徴とする。

【0020】

この請求項6にかかる発明によれば、リソース使用情報として、通常起動時に起動アプリケーションが実際に使用したリソース使用量の平均値を求めることで、実体に則したリソース使用量が得られるようになり、アプリケーションの起動を必要以上に制限したり、逆に、リソース不足による画像形成装置の動作停止を防止することができる。

【0021】

また、請求項7にかかる発明は、請求項2～6のいずれか一つに記載の画像形成装置において、前記リソース使用情報取得手段は、前記起動アプリケーションが使用するメモリ領域情報を取得して前記リソース使用情報を生成し、前記アプリ起動制限手段は、画像形成装置におけるメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するメモリ使用量を除いたメモリ領域残容量を前記リソース量情報として取得し、取得したメモリ領域残容量と、前記メモリ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする。

【0022】

この請求項7にかかる発明によれば、リソース使用情報取得手段によって、起動アプリケーションが使用するメモリ領域情報を取得してリソース使用情報を生成し、アプリ起動制限手段によって、画像形成装置のメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するメモリ使用量を除いたメモリ領域残容量をリソース量情報として取得し、その取得したメモリ領域残容量とメモリ領域情報とに基づいて、起動アプリケーションの起動を制限することで、アプリケーションの実行に最も影響を及ぼすメモリ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができる。

【0023】

また、請求項 8 にかかる発明は、請求項 7 に記載の画像形成装置において、前記リソース使用情報取得手段は、前記起動アプリケーションが使用するテキストメモリ領域情報を取得して前記リソース使用情報を生成し、前記アプリ起動制限手段は、画像形成装置におけるテキストメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するテキストメモリ使用量を除いたテキストメモリ領域残容量を前記リソース量情報として取得し、取得したテキストメモリ領域残容量と、前記テキストメモリ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする。

【0024】

この請求項 8 にかかる発明によれば、リソース使用情報取得手段によって、起動アプリケーションが使用するテキストメモリ領域情報を取得してリソース使用情報を生成し、アプリ起動制限手段によって、画像形成装置におけるテキストメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するテキストメモリ使用量を除いたテキストメモリ領域残容量をリソース量情報として取得し、取得したテキストメモリ領域残容量とテキストメモリ領域情報とに基づいて、起動アプリケーションの起動を制限することで、アプリケーションのプログラムなどを格納するテキストメモリ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができる。

【0025】

また、請求項 9 にかかる発明は、請求項 7 または 8 に記載の画像形成装置において、前記リソース使用情報取得手段は、前記起動アプリケーションが使用するヒープ領域情報を取得して前記リソース使用情報を生成し、前記アプリ起動制限手段は、画像形成装置におけるヒープ領域から画像形成装置特有のアプリケーションおよびシステムが使用するヒープ領域使用量を除いたヒープ領域残容量を前記リソース量情報として取得し、取得したヒープ領域残容量と、前記ヒープ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする。

【0026】

この請求項9にかかる発明によれば、リソース使用情報取得手段によって、起動アプリケーションが使用するヒープ領域情報を取得してリソース使用情報を生成し、アプリ起動制限手段によって、画像形成装置におけるヒープ領域から画像形成装置特有のアプリケーションおよびシステムが使用するヒープ領域使用量を除いたヒープ領域残容量をリソース量情報として取得し、取得したヒープ領域残容量とヒープ領域情報とに基づいて、起動アプリケーションの起動を制限することで、アプリケーションが動的に確保または破棄するヒープ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができる。

【0027】

また、請求項10にかかる発明は、請求項7～9のいずれか一つに記載の画像形成装置において、前記リソース使用情報取得手段は、前記起動アプリケーションが使用するスタック領域情報を取得して前記リソース使用情報を生成し、前記アプリ起動制限手段は、画像形成装置におけるスタック領域から画像形成装置特有のアプリケーションおよびシステムが使用するスタック領域使用量を除いたスタック領域残容量を前記リソース量情報として取得し、取得したスタック領域残容量と、前記スタック領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする。

【0028】

この請求項10にかかる発明によれば、リソース使用情報取得手段によって、起動アプリケーションが使用するスタック領域情報を取得してリソース使用情報を生成し、アプリ起動制限手段によって、画像形成装置におけるスタック領域から画像形成装置特有のアプリケーションおよびシステムが使用するスタック領域使用量を除いたスタック領域残容量をリソース量情報として取得し、取得したスタック領域残容量とスタック領域情報とに基づいて、起動アプリケーションの起動を制限することで、アプリケーションが関数呼び出し、サブルーチン呼び出しなどで使用するスタック領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上さ

せることができる。

【0029】

また、請求項11にかかる発明は、請求項2～10のいずれか一つに記載の画像形成装置において、前記リソース使用情報取得手段は、プロセスが使用するリソースに関する情報を保持するシステム情報から前記起動アプリケーションが使用するリソースに関する情報を取得し、前記リソース使用情報を生成することを特徴とする。

【0030】

この請求項11にかかる発明によれば、リソース使用情報取得手段によって、プロセスが使用するリソースに関する情報を保持するシステム情報からアプリケーションが使用するリソースに関する情報を取得してリソース使用情報を生成することで、システム情報を参照するだけでアプリケーションが使用するリソースの使用情報が取得でき、リソース使用情報取得処理を効率的に行うことができる。

【0031】

また、請求項12にかかる発明は、請求項1～11のいずれか一つに記載の画像形成装置において、前記アプリ起動制限手段は、前記リソース量情報と前記リソース使用情報に基づいて、起動前のアプリケーションに対して起動を制限する起動制限要求メッセージを送信することを特徴とする。

【0032】

この請求項12にかかる発明によれば、アプリ起動制限手段によって、リソース量情報とリソース使用情報に基づいてアプリケーションに対し起動しないように制限を行う起動制限要求メッセージを送信することで、メッセージ送信というプロセス間通信を利用してアプリケーションの起動を制限することができ、多数のプロセスが動作する画像形成装置においてアプリケーションの起動制限を効率的に行うことができる。

【0033】

また、請求項13にかかる発明は、請求項2～12のいずれか一つに記載の画像形成装置において、前記コントロールサービスをサーバとしたクライアントプ

ロセスとして動作し、かつ前記アプリケーションをクライアントとしたサーバプロセスとして動作する仮想アプリケーションサービスをさらに備え、前記リソース使用情報取得手段と前記アプリ起動制限手段は、前記仮想アプリケーションサービスに含まれることを特徴とする。

【0034】

この請求項13にかかる発明によれば、リソース使用情報取得手段と前記アプリ起動制限手段は仮想アプリケーションサービスに含まれており、仮想アプリケーションサービスがコントロールサービスをサーバとしたクライアントプロセスとして動作し、かつアプリケーションをクライアントとしたサーバプロセスとして動作することで、アプリケーションに対してリソース使用情報取得およびアプリ起動制限というサービスを提供するだけでなく、コントロールサービスから画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得することができ、画像形成装置特有のアプリケーションの動作に影響を与えることなく、新規アプリケーションを搭載して利用することができる。

【0035】

また、請求項14にかかる発明は、請求項13に記載の画像形成装置において、前記アプリ起動制限手段と前記リソース使用情報取得手段は、前記仮想アプリケーションサービスのプロセス内部でスレッドとして生成されることを特徴とする。

【0036】

この請求項14にかかる発明によれば、アプリ起動制限手段とリソース使用情報取得手段が仮想アプリケーションサービスプロセス内のスレッドとして生成されるため、複数のアプリケーションが並列に起動された場合に、リソース使用情報取得処理とアプリ起動制限処理とをコンテキストの切り替えなしにCPUの占有切り替えによって並列実行することが可能となり、リソース使用情報取得処理とアプリ起動制限処理との並列処理を円滑に行うことができる。

【0037】

また、請求項15にかかる発明は、請求項13または14に記載の画像形成装

置において、前記仮想アプリケーションサービスは、前記アプリケーション毎に設けられ、前記画像形成装置に特有のアプリケーション以外のアプリケーションを起動する前に、対応する仮想アプリケーションサービスのリソース使用情報取得手段とアプリ起動制限手段とを使って、画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報と、起動するアプリケーションのリソース使用量に関するリソース使用情報とを取得し、リソース使用量がリソース量を越えている場合にアプリケーションの起動を制限することを特徴とする。

【0038】

この請求項15にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーションを起動する前に、アプリケーション毎に設けられた対応する仮想アプリケーションサービスのリソース使用情報取得手段とアプリ起動制限手段とにより、起動するアプリケーションのリソース使用量と、利用可能なリソース量とを取得し、リソース使用量がリソース量を越えている場合にアプリケーションの起動を制限することで、リソース不足によりアプリケーションの実行ができなくなることを未然に回避し、画像形成装置のシステムの安定性を向上させることができる。また、仮想アプリケーションサービスは、各アプリケーション毎に設けられ、アプリケーションごとに別個に処理可能なため、複数のアプリケーションの起動制限処理を各アプリケーションに対応する仮想アプリケーションサービスで並列に実行することにより、アプリケーションの起動制限処理を効率的に行うことができる。

【0039】

また、請求項16にかかる発明は、請求項1～15のいずれか一つに記載の画像形成装置において、前記コントロールサービスは、前記ユーザサービスを提供する際に、アプリケーションの少なくとも2つが共通的に必要とする前記ハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うことを特徴とする。

【0040】

この請求項16にかかる発明によれば、コントロールサービスは、ユーザサー

ビスを提供する際に、アプリケーションの少なくとも 2 つが共通的に必要とするハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うようにすることで、複数のアプリケーションを作成する際に、共通化できる部分をコントロールサービス（プラットフォーム）として共通化することにより、それ以外の部分を作成すればよいので、アプリケーションの開発が容易となり、かつ、短期間で開発することができる。

【0041】

また、請求項 17 にかかる発明は、画像形成処理にかかるユーザサービスにそれぞれ固有の処理を行うアプリケーションと、前記アプリケーションと画像形成処理で使用するハードウェア資源との間に介在し、前記ハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うコントロールサービスとを備えた画像形成装置上で、前記アプリケーションの起動制限を行うアプリ起動制限方法であって、前記画像形成装置に特有のアプリケーション以外のアプリケーションを起動する前に、前記画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得し、取得したリソース量情報と、上記起動アプリケーションが使用するリソース使用量に関するリソース使用情報とに基づいて、前記リソース使用量が前記リソース量を越えている場合は前記起動アプリケーションの起動を制限するアプリ起動制限ステップを含むことを特徴とする。

【0042】

この請求項 17 にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーションが起動される前に、アプリ起動制限ステップによって、画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得し、取得したリソース量情報と、起動アプリケーションが使用するリソース使用量に関するリソース使用情報とに基づいて、リソース使用量がリソース量を越えている場合に起動アプリケーションの起動を制限することで、リソースの状況によって画像形成装置特有のアプリケーションの実行ができなくなることを未然に回避し、画像形成装置のシステムの安定性を向上させることができる。

【 0 0 4 3 】

また、請求項 1 8 にかかる発明は、請求項 1 7 に記載のアプリ起動制限方法において、前記画像形成装置に特有のアプリケーション以外の起動アプリケーションが使用するリソース使用量に関する情報を取得して前記リソース使用情報を生成するリソース使用情報取得ステップをさらに含むことを特徴とする。

【 0 0 4 4 】

この請求項 1 8 にかかる発明によれば、リソース使用情報取得ステップによって、画像形成装置に特有のアプリケーション以外の起動アプリケーションが使用するリソース使用量に関する情報を取得してリソース使用情報を生成することで、起動アプリケーションごとに実際に必要とするリソースに関するリソース使用情報に基づいてアプリ起動制限ステップによりアプリ起動を制限することができ、画像形成装置のシステムの安定性をより向上させることができる。

【 0 0 4 5 】

また、請求項 1 9 にかかる発明は、請求項 1 8 に記載のアプリ起動制限方法において、前記リソース使用情報取得ステップは、前記起動アプリケーションが自ら保有するリソース使用量に関する情報を取得して前記リソース使用情報を生成することを特徴とする。

【 0 0 4 6 】

この請求項 1 9 にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーション（ユーザやサードベンダが開発した新規アプリケーションなど）を起動する前に、リソース使用情報取得ステップによって、各アプリケーションが自ら保有するリソース使用量に関する情報を取得し、リソース使用情報を生成することで、起動アプリケーションが使用する正確なリソース使用量を事前に把握することが可能となり、制限すべきアプリケーションの起動か否かを判断することができる。

【 0 0 4 7 】

また、請求項 2 0 にかかる発明は、請求項 1 8 に記載のアプリ起動制限方法において、前記リソース使用情報取得ステップは、前記起動アプリケーションを仮起動することにより、当該起動アプリケーションが使用するリソース使用量に関

する情報を取得して前記リソース使用情報を生成することを特徴とする。

【0048】

この請求項 20 にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーション（ユーザやサードベンダが開発した新規アプリケーションなど）を通常起動する前に、リソース使用情報取得ステップによって、起動前のアプリケーションを仮起動させることにより、そのアプリケーションが使用するリソース使用量に関する情報を取得することができ、ここからリソース使用情報を生成することで、起動アプリケーションが使用する正確なリソース使用量を事前に把握することが可能となり、制限すべきアプリケーションの起動か否かを判断することができる。

【0049】

また、請求項 21 にかかる発明は、請求項 19 または 20 に記載のアプリ起動制限方法において、前記リソース使用情報取得ステップは、前記起動アプリケーションが自ら保有するリソース使用量、あるいは、前記起動アプリケーションの仮起動によるリソース使用量に基づいて通常起動を行い、その通常起動時に当該起動アプリケーションが実際に使用するリソース使用量に基づいて前記リソース使用情報を生成することを特徴とする。

【0050】

この請求項 21 にかかる発明によれば、リソース使用情報取得ステップによって、起動アプリケーションが自ら保有するリソース使用量、あるいは、起動アプリケーションの仮起動によるリソース使用量に基づいて通常起動を行い、通常起動時に起動アプリケーションが実際に使用したリソース使用量に基づいてリソース使用情報を生成することで、起動アプリケーションが実際に使用するより正確なリソース使用量が把握できるようになり、当該アプリケーションが再度起動される前に、制限すべきアプリケーションの起動か否かを判断することができる。特に、単独でのアプリケーションの起動では問題にならない程度のリソース使用量であっても、複数のアプリケーションを同時または追加的に起動する場合は、それらの合計リソース使用量が問題となるため、各アプリケーションが使用するリソース使用量をできるだけ正確に把握することが必要となる。

【0051】

また、請求項 22 にかかる発明は、請求項 21 に記載のアプリ起動制限方法において、前記リソース使用情報は、前記起動アプリケーションの通常起動時に実際に使用されるリソース量の平均値としたことを特徴とする。

【0052】

この請求項 22 にかかる発明によれば、リソース使用情報として、通常起動時に起動アプリケーションが実際に使用したリソース使用量の平均値を求めることで、実体に則したリソース使用量が得られるようになり、アプリケーションの起動を必要以上に制限したり、逆に、リソース不足による画像形成装置の動作停止を防止することができる。

【0053】

また、請求項 23 にかかる発明は、請求項 18～22 のいずれか一つに記載のアプリ起動制限方法において、前記リソース使用情報取得ステップは、前記起動アプリケーションが使用するメモリ領域情報を取得して前記リソース使用情報を生成し、前記アプリ起動制限ステップは、画像形成装置におけるメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するメモリ使用量を除いたメモリ領域残容量を前記リソース量情報として取得し、取得したメモリ領域残容量と、前記メモリ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする。

【0054】

この請求項 23 にかかる発明によれば、リソース使用情報取得ステップによって、起動アプリケーションが使用するメモリ領域情報を取得してリソース使用情報を生成し、アプリ起動制限ステップによって、画像形成装置のメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するメモリ使用量を除いたメモリ領域残容量をリソース量情報として取得し、その取得したメモリ領域残容量とメモリ領域情報とに基づいて、起動アプリケーションの起動を制限することで、アプリケーションの実行に最も影響を及ぼすメモリ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができる。

【0055】

また、請求項 24 にかかる発明は、請求項 23 に記載のアプリ起動制限方法において、前記リソース使用情報取得ステップは、前記起動アプリケーションが使用するテキストメモリ領域情報を取得して前記リソース使用情報を生成し、前記アプリ起動制限ステップは、画像形成装置におけるテキストメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するテキストメモリ使用量を除いたテキストメモリ領域残容量を前記リソース量情報として取得し、取得したテキストメモリ領域残容量と、前記テキストメモリ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする。

【0056】

この請求項 24 にかかる発明によれば、リソース使用情報取得ステップによって、起動アプリケーションが使用するテキストメモリ領域情報を取得してリソース使用情報を生成し、アプリ起動制限ステップによって、画像形成装置におけるテキストメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するテキストメモリ使用量を除いたテキストメモリ領域残容量をリソース量情報として取得し、取得したテキストメモリ領域残容量とテキストメモリ領域情報とに基づいて、起動アプリケーションの起動を制限することで、アプリケーションのプログラムなどを格納するテキストメモリ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができる。

【0057】

また、請求項 25 にかかる発明は、請求項 23 または 24 に記載のアプリ起動制限方法において、前記リソース使用情報取得ステップは、前記起動アプリケーションが使用するヒープ領域情報を取得して前記リソース使用情報を生成し、前記アプリ起動制限ステップは、画像形成装置におけるヒープ領域から画像形成装置特有のアプリケーションおよびシステムが使用するヒープ領域使用量を除いたヒープ領域残容量を前記リソース量情報として取得し、取得したヒープ領域残容量と、前記ヒープ領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする。

【0058】

この請求項 25 にかかる発明によれば、リソース使用情報取得ステップによって、起動アプリケーションが使用するヒープ領域情報を取得してリソース使用情報を生成し、アプリ起動制限ステップによって、画像形成装置におけるヒープ領域から画像形成装置特有のアプリケーションおよびシステムが使用するヒープ領域使用量を除いたヒープ領域残容量をリソース量情報として取得し、取得したヒープ領域残容量とヒープ領域情報とに基づいて、起動アプリケーションの起動を制限することで、アプリケーションが動的に確保または破棄するヒープ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができる。

【0059】

また、請求項 26 にかかる発明は、請求項 23～25 のいずれか一つに記載のアプリ起動制限方法において、前記リソース使用情報取得ステップは、前記起動アプリケーションが使用するスタック領域情報を取得して前記リソース使用情報を生成し、前記アプリ起動制限ステップは、画像形成装置におけるスタック領域から画像形成装置特有のアプリケーションおよびシステムが使用するスタック領域使用量を除いたスタック領域残容量を前記リソース量情報として取得し、取得したスタック領域残容量と、前記スタック領域情報とに基づいて、前記起動アプリケーションの起動を制限することを特徴とする。

【0060】

この請求項 26 にかかる発明によれば、リソース使用情報取得ステップによって、起動アプリケーションが使用するスタック領域情報を取得してリソース使用情報を生成し、アプリ起動制限ステップによって、画像形成装置におけるスタック領域から画像形成装置特有のアプリケーションおよびシステムが使用するスタック領域使用量を除いたスタック領域残容量をリソース量情報として取得し、取得したスタック領域残容量とスタック領域情報とに基づいて、起動アプリケーションの起動を制限することで、アプリケーションが関数呼び出し、サブルーチン呼び出しなどで使用するスタック領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性

を向上させることができる。

【0061】

また、請求項 27 にかかる発明は、請求項 18～26 のいずれか一つに記載のアプリ起動制限方法において、前記リソース使用情報取得ステップは、プロセスが使用するリソースに関する情報を保持するシステム情報から前記起動アプリケーションが使用するリソースに関する情報を取得し、前記リソース使用情報を生成することを特徴とする。

【0062】

この請求項 27 にかかる発明によれば、リソース使用情報取得ステップによって、プロセスが使用するリソースに関する情報を保持するシステム情報からアプリケーションが使用するリソースに関する情報を取得してリソース使用情報を生成することで、システム情報を参照するだけでアプリケーションが使用するリソースの使用情報が取得でき、リソース使用情報取得処理を効率的に行うことができる。

【0063】

また、請求項 28 にかかる発明は、請求項 17～27 のいずれか一つに記載のアプリ起動制限方法において、前記アプリ起動制限ステップは、前記リソース量情報と前記リソース使用情報に基づいて、起動前のアプリケーションに対して起動を制限する起動制限要求メッセージを送信することを特徴とする。

【0064】

この請求項 28 にかかる発明によれば、アプリ起動制限ステップによって、リソース量情報とリソース使用情報に基づいてアプリケーションに対し起動しないように制限を行う起動制限要求メッセージを送信することで、メッセージ送信というプロセス間通信を利用してアプリケーションの起動を制限することができ、多数のプロセスが動作する画像形成装置においてアプリケーションの起動制限を効率的に行うことができる。

【0065】

【発明の実施の形態】

以下に添付図面を参照して、この発明にかかる画像形成装置およびアプリ起動

制限方法の好適な実施の形態を詳細に説明する。

【0066】

(実施の形態1)

図1は、この発明の実施の形態1である画像形成装置（以下、「複合機」という）の構成を示すブロック図である。図1に示すように、複合機100は、白黒レーザプリンタ（B&W LP）101と、カラーレーザプリンタ（Color LP）102と、スキャナ、ファクシミリ、ハードディスク、メモリ、ネットワークインタフェースなどのハードウェアリソース103を有するとともに、プラットフォーム120とアプリケーション（以下、アプリともいう）130と仮想アプリケーションサービス（VAS：Virtual Application Service）140から構成されるソフトウェア群110とを備えている。

【0067】

一般のパーソナルコンピュータのユーザの場合は、多数のアプリを順次起動していくと、リソース不足が生じてストールする可能性があることを承知の上で利用しているが、本発明のような複合機のユーザの場合は、複合機に特有のコピー、プリンタ、スキャナ、ファクシミリといったアプリに関しては、いつでも起動して利用できるのが当然であって、複合機特有のアプリを起動させた時にストールして利用できなくなることは想定していない。しかし、ユーザやサードベンダなどが開発した新規アプリが搭載可能な複合機の場合、リソース使用量が多い複数の新規アプリの起動中に、複合機特有のアプリを起動させるとリソース不足が生じるおそれがあり、ユーザの想定していなかった複合機のストールが現実のものとなりつつある。そこで、本発明の特徴的な構成の一つである仮想アプリケーションサービス（VAS）140は、アプリ130とプラットフォーム120との間に配置されている。このVAS140は、アプリ130がインストールされた後、アプリを起動する前に、アプリの使用するリソース使用量に関するリソース使用情報を取得して、リソース使用情報ファイルをハードディスク（HD）に生成する（後述する図2のHD200を参照）。取得したリソースとしては、メモリに確保されるテキストメモリ領域サイズ、ヒープ領域サイズ、スタック領域サイズがある。ここで、テキストメモリ領域とは、アプリケーション130の

各アプリのプログラムがロードされるメモリ領域である。ヒープ領域とは、各アプリが動的に確保するメモリ領域である。スタック領域とは、各アプリが実行される時、または各アプリが内部のモジュールを呼び出すときに使用する引数などを格納するために確保される領域である。

【0068】

また、VAS140は、コピー、プリンタ、スキャナ、ファクシミリといった複合機に特有のアプリ以外の新規アプリを起動する前に、複合機特有のアプリおよびシステムが使用するリソース使用量を取得し、複合機の総リソース量からリソース使用量を除いた利用可能なリソース量情報を取得する。例えば、複合機の総リソース量は、あらかじめ分かっているため、ハードディスク（HD）のリソース使用情報ファイルを参照して、複合機特有のアプリおよびシステムが使用するリソース使用量から利用可能なリソース量情報を得ることができる。また、複合機特有のアプリであれば、そのアプリおよびシステムが使用するリソース使用量も既知であるため、複合機の総リソース量から差し引いた利用可能なリソース量情報は、既知の情報として利用することができる。

【0069】

また、VAS140のリソース使用情報の取得方法としては、以下の①～③などを用いている。①各アプリの使用するリソース使用量に関する情報（リソース使用情報）をアプリ自身が保有している場合は、アプリからの自己申告によってリソース使用情報を得ることができる。②通常起動前のアプリをVASが仮起動させることにより、当該アプリが使用するリソース量に関する情報（リソース使用情報）を得ることができる。③上記の①または②で得たリソース使用情報に基づいてアプリを正常に起動させ、そのアプリの起動中に実際に使用したリソース使用量を収集することで、次回以降の通常起動時に使用する正確なリソース使用情報を得ることができる。

【0070】

そして、VAS140は、複合機特有のアプリ以外の新規アプリの起動前に、利用可能なリソース量情報と、起動アプリが使用するリソース使用量に関するリソース使用情報とを取得し、両者を比較することにより、複合機特有のアプリを

起動した場合にリソース不足によってストールする可能性のある新規アプリの起動を制限する。具体的には、起動アプリのリソース使用量が利用可能なリソース量を越えている場合、起動前の新規アプリに対して起動制限要求メッセージを送信することにより、新規アプリの起動を制限する。

【0071】

プラットフォーム120は、アプリケーションからの処理要求を解釈してハードウェア資源の獲得要求を発生させるコントロールサービスと、一または複数のハードウェア資源の管理を行い、コントロールサービスからの獲得要求を調停するシステムリソースマネージャ（SRM）123と、汎用OS121とを有している。

【0072】

コントロールサービスは、複数のサービスモジュールから形成され、SCS（システムコントロールサービス）122と、ECS（エンジンコントロールサービス）124と、MCS（メモリコントロールサービス）125と、OCS（オペレーションパネルコントロールサービス）126と、FCS（ファックスコントロールサービス）127と、NCS（ネットワークコントロールサービス）128とから構成されている。なお、このプラットフォーム120は、あらかじめ定義された関数により前記アプリケーション130から処理要求を受信可能とするアプリケーションプログラムインタフェース（API）を有している。

【0073】

汎用OS121は、UNIX（登録商標）などの汎用オペレーティングシステムであり、プラットフォーム120並びにアプリケーション130の各ソフトウェアをそれぞれプロセスとして並列実行する。

【0074】

SRM123のプロセスは、SCS122とともにシステムの制御およびリソースの管理を行うものである。SRM123のプロセスは、スキャナ部やプリンタ部などのエンジン、メモリ、HDDファイル、ホストI/O（セントロI/F、ネットワークI/F、IEEE1394 I/F、RS232C I/Fなど）のハードウェア資源を利用する上位層からの要求にしたがって調停を行い、実

行制御する。

【 0 0 7 5 】

具体的には、この S R M 1 2 3 は、要求されたハードウェア資源が利用可能であるか（他の要求により利用されていないかどうか）を判断し、利用可能であれば要求されたハードウェア資源が利用可能である旨を上位層に伝える。また、S R M 1 2 3 は、上位層からの要求に対してハードウェア資源の利用スケジューリングを行い、要求内容（例えば、プリンタエンジンにより紙搬送と作像動作、メモリ確保、ファイル生成など）を直接実施している。

【 0 0 7 6 】

S C S 1 2 2 のプロセスは、アプリ管理、操作部制御、システム画面表示、L E D 表示、リソース管理、割り込みアプリ制御などを行う。

【 0 0 7 7 】

E C S 1 2 4 のプロセスは、白黒レーザプリンタ（B&W LP） 1 0 1、カラーレーザプリンタ（Color LP） 1 0 2、スキャナ、ファクシミリなどからなるハードウェアリソース 1 0 3 のエンジンの制御を行う。

【 0 0 7 8 】

M C S 1 2 5 のプロセスは、画像メモリの取得および解放、ハードディスク装置（HDD）の利用、画像データの圧縮および伸張などを行う。

【 0 0 7 9 】

F C S 1 2 7 のプロセスは、システムコントローラの各アプリ層から P S T N / I S D N 網を利用したファクシミリ送受信、B K M（バックアップ S R A M）で管理されている各種ファクシミリデータの登録／引用、ファクシミリ読みとり、ファクシミリ受信印刷、融合送受信を行うための A P I を提供する。

【 0 0 8 0 】

N C S 1 2 8 のプロセスは、ネットワーク I / O を必要とするアプリケーションに対して共通に利用できるサービスを提供するためのプロセスであり、ネットワーク側から各プロトコルによって受信したデータを各アプリケーションに振り分けたり、アプリケーションからデータをネットワーク側に送信する際の仲介を行う。具体的には、ftpd、httpd、lpd、snmpd、telnetd、smtpdなどのサーバデ

ーモンや、同プロトコルのクライアント機能などを有している。

【0081】

OCS126のプロセスは、オペレータ（ユーザ）と本体制御間の情報伝達手段となるオペレーションパネル（操作パネル）の制御を行う。OCS126は、オペレーションパネルからキー押下をキーイベントとして取得し、取得したキーに対応したキーイベント関数をSCS122に送信するOCSプロセスの部分と、アプリケーション130またはコントロールサービスからの要求によりオペレーションパネルに各種画面を描画出力する描画関数やその他オペレーションパネルに対する制御を行う関数などがあらかじめ登録されたOCSライブラリの部分とから構成される。このOCSライブラリは、アプリケーション130およびコントロールサービスの各モジュールにリンクされて実装されている。なお、OCS126のすべてをプロセスとして動作させるように構成しても良く、あるいはOCS126のすべてをOCSライブラリとして構成しても良い。

【0082】

アプリケーション130は、ページ記述言語（PDL）、PCLおよびポストスクリプト（PS）を有するプリンタ用のアプリケーションであるプリンタアプリ111と、コピー用アプリケーションであるコピーアプリ112と、ファクシミリ用アプリケーションであるファックスアプリ113と、スキャナ用アプリケーションであるスキャナアプリ114と、ネットワークファイル用アプリケーションであるネットファイルアプリ115と、工程検査用アプリケーションである工程検査アプリ116とを有している。これらの各アプリは、複合機特有（画像形成装置特有）のアプリであり、起動時にVAS140に対して自プロセスのプロセスIDとともにアプリ登録要求メッセージを送信し、アプリ登録要求メッセージを受信したVAS140によって、起動したアプリに対する登録処理が行われるようになっている。

【0083】

アプリケーション130の各プロセス、コントロールサービスの各プロセスは、関数呼び出しとその戻り値送信およびメッセージの送受信によってプロセス間通信を行いながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成

処理にかかるユーザサービスを実現している。

【0084】

このように、実施の形態1にかかる複合機100には、複数のアプリケーション130および複数のコントロールサービスが存在し、いずれもプロセスとして動作している。そして、これらの各プロセス内部には、一または複数のスレッドが生成されて、スレッド単位の並列実行が行われる。そして、コントロールサービスがアプリケーション130に対し共通サービスを提供しており、このため、これらの多数のプロセスが並列動作、およびスレッドの並列動作を行って互いにプロセス間通信を行って協調動作をしながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成処理にかかるユーザサービスを提供するようになっていいる。また、複合機100には、サードベンダなどの第三者がコントロールサービス層の上のアプリケーション層に新規アプリ117, 118を開発して搭載することが可能となっている。図1では、この新規アプリ117, 118を搭載した例を示している。この新規アプリ117, 118は、複合機特有のアプリ以外のアプリに相当するものである。

【0085】

なお、実施の形態1にかかる複合機100では、複数のアプリケーション130のプロセスと複数のコントロールサービスのプロセスとが動作しているが、アプリケーション130とコントロールサービスのプロセスがそれぞれ単一の構成とすることも可能である。また、各アプリケーション130は、アプリケーションごとに追加または削除することができる。

【0086】

図2は、実施の形態1にかかる複合機100のVAS140の構成と、VAS140と各アプリ、コントロールサービス層150および汎用OS121との関係を示すブロック図である。なお、図2では、アプリケーション130の例として、プリンタアプリ111、コピーアプリ112、新規アプリ117, 118を示しているが、他のアプリでも同様の構成である。

【0087】

仮想アプリケーションサービス（VAS）140のプロセスには、ディスパッ

チャ 144 と、制御スレッド 143 と、リソース使用情報取得手段としてのリソース使用情報取得スレッド 141 と、アプリ起動制限手段としてのアプリ起動制限スレッド 142 とが動作している。

【0088】

ディスパッチャ 144 は、アプリケーション 130 やコントロールサービスからのメッセージ受信を監視し、受信したメッセージに応じて制御スレッド 143、リソース使用情報取得スレッド 141、アプリ起動制限スレッド 142 に対して処理要求を行うものである。実施の形態 1 の複合機 100 では、ディスパッチャ 144 は、コントロールサービスから複合機特有のアプリ以外の新規アプリに対するアプリ起動要求メッセージを受信したとき、受信したアプリ起動要求メッセージを制御スレッド 143 に送信し、制御スレッド 143 からリソース使用情報取得スレッド 141 とアプリ起動制限スレッド 142 に対して処理要求を行うようになっている。

【0089】

制御スレッド 143 は、ディスパッチャ 144 からのアプリ起動要求メッセージを受信して、リソース使用情報取得処理およびアプリ起動制限処理を行う。ここで、アプリ起動制限処理とは、必ずアプリ起動制限を行うものではなく、起動するアプリのリソース使用量に応じて、アプリを通常起動させるかアプリを起動制限するかを選択する処理をいう。

【0090】

また、制御スレッド 143 は、HD 200 に格納されたリソース使用情報ファイル 201 を参照することにより、インストールされたアプリの中でリソース使用量に関するリソース使用情報が記録されていないアプリの有無を判別することができる。そして、複合機特有のアプリ以外の新規アプリに対する起動要求がコントロールサービスから有ると、制御スレッド 143 は、リソース使用情報取得スレッド 141 によって起動アプリのリソース使用量を取得するか、HD 200 内のリソース使用情報ファイル 201 からリソース使用情報を読み出して、アプリ起動制限スレッド 142 に送信する。また、リソース使用情報ファイル 201 に格納されている複合機特有のアプリが使用するリソースもアプリ起動制限スレ

ッド142に送られる。アプリ起動制限スレッド142では、複合機の総リソース量から複合機特有のアプリが使用するリソースを除いた利用可能なリソース量情報と、新規アプリが使用するリソース使用量とを比較して、リソース使用量の方が利用可能なリソース量を越えている場合は、起動要求のあったアプリに対して起動制限処理する。

【0091】

リソース使用情報取得スレッド141は、制御スレッド143から処理要求を受けると、汎用OS121が管理するRAM210上のproc構造体211（またはu領域212）を参照して、各アプリが使用するテキストメモリ領域サイズ、ヒープ領域サイズ、スタック領域サイズを取得して、リソース使用情報ファイル201としてハードディスク（HD）200に生成する。かかるリソース使用情報は、アプリごとのレコードとして記録される。

【0092】

上記VAS140のプログラムは、ソフトウェア開発キット（SDK：Software Development Kit）等の一部または全部として、CD-ROMまたはFD（フレキシブルディスク）などの記憶媒体に実行可能な形式またはインストール可能な形式のファイルで提供される。また、このような実行可能な形式またはインストール可能な形式のVAS140のプログラムファイルを、ネットワーク経由で取得可能な方法で提供するようにしても良い。

【0093】

図3は、HD200に格納されるリソース使用情報ファイル201の内容例を示す説明図である。図3に示すように、リソース使用情報ファイル201には、アプリIDごとに、テキストメモリ領域サイズ、ヒープ領域サイズ、スタック領域サイズなどが記録されている。

【0094】

図4は、リソース使用情報取得スレッド141が参照するproc構造体211の一例を示す説明図である。図4に示すように、proc構造体211には、各プロセスごとに、プロセスID（p_pid）、テキストメモリ領域サイズ、ヒープメモリ領域サイズ、スタック領域サイズなどが格納されている。このpr

o c 構造体 211 は、プロセス実行時、プロセス終了時、およびプロセスの状態が変化したときに汎用 OS 121 によって更新される。

【0095】

アプリ起動制限スレッド 142 は、コントロールサービス層 150 から新規アプリの起動要求があった場合に、リソース使用情報ファイル 201 を参照して、その新規アプリが使用するリソース使用量に関する情報を取得する。また、アプリ起動制限スレッド 142 は、汎用 OS 121 のシステムコールあるいはコントロールサービスで提供されるサービス関数呼び出しによって、複合機特有のアプリやシステムが使用するリソースを除いた利用可能な残りソース量（例えば、画像メモリ領域の残容量、ヒープ領域の残容量、スタック領域など）を取得する。そして、両者を比較して、アプリ起動を制限する必要があるか否かを判断する。

【0096】

アプリ起動制限スレッド 142 は、アプリ起動制限が必要と判断すると、アプリ起動要求のあったアプリに対して起動制限要求メッセージを送信する。一方、アプリ起動制限をすべきでないと判断した場合は、アプリ起動要求のあったアプリに対して起動要求メッセージを送信して、アプリの通常起動処理が行われる。

【0097】

図 5 は、複合機のオペレーションパネル上にユーザアプリの起動制限レベルをヒストグラムで表示した図である。図 5 に表示されるヒストグラムは、複合機特有のアプリやシステムが使用するリソースを除いた利用可能な残りソース量のみを示したものである。すなわち、残りソースを複数のユーザアプリ（新規アプリ）でどのように分配して利用すれば良いかをユーザに視覚で判断できるようにしたものである。ヒストグラムの内訳は、1 番目に起動したユーザアプリ 221 のリソース使用量と、2 番目に起動したユーザアプリ 222 のリソース使用量とがそれぞれ表示され、これから起動しようとする 3 番目のユーザアプリ 223 の場合は、予想リソース使用量を示している。図 5 の例では、3 番目のユーザアプリ 223 を起動すると残りソース量が危険レベルに達するので、起動制限を受ける可能性があることがわかる。ここでは、残りソース量をヒストグラムで表すだけでなく、起動制限レベルを安全（青）、注意（黄）、危険（赤）の 3 段階に色分

け表示して、さらに視認効果を向上させている。なお、これ以外にも、危険レベルになると表示を点滅させたり、音声ガイダンスや警告音を発するようにしても勿論良い。

【0098】

次に、このように構成された複合機100のVAS140によるアプリ起動制限処理について説明する。図6は、VAS140の制御スレッド143により起動要求のあったアプリ起動制限処理手順を示すフローチャートである。

【0099】

まず、ディスパッチャ144がコントロールサービス層150からアプリ起動要求メッセージを受信すると、アプリ起動要求メッセージをそのアプリのプロセスIDとともに制御スレッド143に受け渡す。制御スレッド143は、アプリ起動要求メッセージとプロセスIDとをディスパッチャ144から受信すると、起動要求のあったアプリが複合機特有のアプリか否かを判断する（ステップS601）。複合機特有のアプリ以外のアプリの場合、すなわち、新規アプリに対する起動要求があった場合（ステップS602）、制御スレッド143は、そのアプリのリソース使用情報ファイルが作成されているか否かをHD200を参照して判断する（ステップS603）。

【0100】

リソース使用情報ファイルが作成されている場合、制御スレッド143は、起動要求のあった新規アプリのリソース使用情報をリソース使用情報ファイル201からアプリ起動制限スレッド142へ送る（ステップS604）。また、上記ステップS603で、新規アプリのリソース使用情報ファイルが作成されていない場合は、制御スレッド143は、リソース使用情報取得スレッド141に対してリソース使用情報ファイルを作成するように処理要求する（ステップS605）。ステップS605のリソース使用情報ファイルの作成処理の詳細なサブルーチンは、後述する図7のフローチャートを用いて説明する。このリソース使用情報ファイルが作成処理されると、上記したステップS604の処理が行われる。

【0101】

続いて、制御スレッド143は、アプリ起動制限スレッド142に対して、複合機特有のアプリおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得するように要求する。そして、アプリ起動制限スレッド142は、汎用OS121のシステムコールあるいはコントロールサービス層150で提供されるサービス関数呼び出しを用いることにより、複合機特有のアプリやシステムが使用するリソースを除いた利用可能な残リソース量（画像メモリ領域の残容量、ヒープ領域の残容量、スタック領域など）を取得する（ステップS606）。

【0102】

また、アプリ起動制限スレッド142では、リソース使用情報ファイル201を参照して取得した新規アプリが使用するリソース使用量に関する情報と、サービス関数呼び出しによって取得した複合機特有のアプリやシステムが使用するリソースを除いた利用可能なリソース量情報とを比較する（ステップS607）。この比較結果は、VAS140の制御スレッド143からの要求によって、不図示のオペレーションパネルに対して各種画面を描画出力する描画関数やその他の制御を行う関数などがあらかじめ登録されたOCSライブラリを用いることにより、図5に示したようなユーザアプリの起動制限レベルを表示するヒストグラムを描画出力する（ステップS608）。

【0103】

ここで、アプリ起動制限スレッド142は、リソース使用量がリソース量を越えているか否かを判断し（ステップS609）、越えている場合は、後から複合機特有のアプリが起動されるとリソース不足が生じて動作が不安定となったり、停止する事態を招くので、起動要求のあった新規アプリに対して起動制限処理する（ステップS610）。

【0104】

また、アプリ起動制限スレッド142は、ステップS609においてリソース使用量がリソース量を越えていない場合、アプリ起動制限処理は行わず、通常のアプリ起動処理が行われる（ステップS611）。このステップS611の通常のアプリ起動処理は、図6の最初のステップS601において、複合機特有のア

プリに対する起動要求があった場合も同様に行われる。本発明において、複合機特有のアプリの場合は、使用するリソースがあらかじめ確保されているため、何時起動処理を行ってもリソース不足が生じないようにになっている。

【0105】

図7は、リソース使用情報ファイルの作成処理の動作を説明するフローチャートである。なお、この図7の動作は、図6のステップS605のサブルーチンとして行っているが、各アプリがインストールされてから起動要求される前までに、リソース使用情報ファイルを作成しても良い。

【0106】

まず、図7において、制御スレッド143は、アプリのインストールの有無を判断し（ステップS701）、インストールされたアプリが有り、そのアプリが自ら保有するリソース使用情報の自己申告が有ると（ステップS702）、リソース使用情報取得スレッド141に対してリソース使用情報ファイルの作成処理を要求する（ステップS703）。

【0107】

リソース使用情報取得スレッド141は、自己申告のあったアプリのリソース使用情報に基づいて、HD200内のリソース使用情報ファイル201に各アプリ単位でリソース使用情報ファイルを作成する（ステップS704）。図3は、作成されたリソース使用情報ファイルの一例を示したものであるが、アプリIDごとに、その使用リソースとして、テキストメモリ領域サイズ、ヒープ領域サイズ、スタック領域サイズごとにリソース量が格納されている。

【0108】

また、上記ステップS702において、アプリからリソース使用情報の自己申告が無かった場合は、リソース使用情報を得たいアプリに対してVAS140が仮起動させることにより（ステップS705）、大まかなリソース使用情報を取得することができる。リソース使用情報を取得した制御スレッド143は、リソース使用情報取得スレッド141に対してリソース使用情報ファイルの作成処理を要求して（ステップS703）、HD200内にリソース使用情報ファイル201を作成する（ステップS704）。

【0109】

さらに、図7のステップS702の自己申告、あるいは、ステップS705のアプリの仮起動により取得したリソース使用情報に基づいて、アプリが正常に起動処理された場合は（図6のステップS611参照）、当該アプリが通常起動中に実際に使用するリソース使用情報を取得して記録することができる。これによって得られるリソース使用情報は、汎用OS121が管理するRAM210上のproc構造体211（またはu領域212）に記録するようにする。そして、このリソース使用情報をアプリが起動されるたびに更新すれば、学習機能を持たせたのと同様に常に正確なリソース使用情報を得ることができる。なお、取得したリソース使用情報のリソースが変動する場合は、ここではその平均値、あるいは、最大値（MAX）をリソース使用情報としている。

【0110】

このRAM210のproc構造体211のリソース使用情報を使ってリソース使用情報ファイルを作成する場合は、リソース使用情報取得スレッド141がアプリID、プロセスIDとリソース使用情報取得要求メッセージを制御スレッド143から受信すると、RAM210のproc構造体211を参照して該当するアプリIDのプロセスIDのブロックの位置を検索する。そして、検索されたプロセスIDのブロックから、テキストメモリ領域サイズ、ヒープ領域サイズ、スタック領域サイズのリソース使用情報を取得し、アプリIDとともにリソース使用情報ファイル201に記録するようにする。

【0111】

このように、実施の形態1にかかる複合機100では、仮想アプリケーションサービス140のリソース使用情報取得スレッド141によって、アプリ130が使用するリソース使用量に関する情報を取得してリソース使用情報ファイル201を生成し、アプリ起動制限スレッド142によって、複合機特有のアプリやシステムが使用するリソースを除いた利用可能なリソース量と、起動要求された新規アプリのリソース使用量とを比較して、リソース使用量がリソース量を越えている場合は、新規アプリの起動を制限するので、リソース不足により動作が不安定になったり、ストールする心配がなくなり、ユーザやサードベンダが開発し

た新規アプリを自由に搭載して起動させることができる。

【0 1 1 2】

また、実施の形態 1 にかかる複合機 1 0 0 では、新規アプリのリソース使用状況によって複合機特有のアプリの実行ができなくなることを未然に回避して、複合機 1 0 0 のシステムの安定性を向上させることができる。

【0 1 1 3】

なお、実施の形態 1 にかかる複合機 1 0 0 では、V A S 1 4 0 が全てのアプリケーション 1 3 0 に対してリソース使用情報取得処理、アプリ起動制限処理を行っているが、一部のアプリに対してのみかかる処理を行うように構成しても良い。例えば、新規アプリ 1 1 7, 1 1 8 などサードベンダなどの第三者が開発したアプリにのみリソース使用情報取得処理、アプリ起動制限処理を行い、プリンタアプリ 1 1 1 やコピーアプリ 1 1 2 などの既存のアプリに対してはこのようなサービスを行わないように構成しても良い。

【0 1 1 4】

また、実施の形態 1 にかかる複合機 1 0 0 では、アプリ起動制限スレッド 1 4 2 により、利用可能なリソース量と新規アプリのリソース使用量とを比較して、リソース使用量がリソース量を越えている場合に新規アプリの起動を制限するようにしたが、新規アプリのリソース使用量の変動することを考えると、リソース使用量がリソース量を越えないまでも、一定の範囲内にある場合は、新規アプリの起動を制限することも状況に応じて考慮する必要がある。

【0 1 1 5】

(実施の形態 2)

実施の形態 1 にかかる複合機 1 0 0 は、V A S 1 4 0 が全アプリケーションに対して 1 つのみ存在するものであったが、この実施の形態 2 にかかる複合機では、各アプリごとに一つの V A S が起動し、各 V A S は対応するアプリに対してのみリソース使用情報取得およびアプリ起動制限を行うものである。

【0 1 1 6】

図 8 は、実施の形態 2 にかかる複合機 8 0 0 の構成を示すブロック図である。図 8 に示すように、複合機 8 0 0 では、複数の仮想アプリケーションサービス (

VAS) 841～848がアプリケーション130の各アプリごとに動作している点が、実施の形態1にかかる複合機100と異なっている。

【0117】

VAS 841～848は、プリンタアプリ111、コピーアプリ112、ファックスアプリ113、スキャナアプリ114、ネットファイルアプリ115、工程検査アプリ116、新規アプリ117および118に対応して、リソース使用情報取得処理およびアプリ起動制限処理を行うようになっている。

【0118】

図9は、実施の形態2にかかる複合機800のVAS 841～848の構成と、VAS 841～848と各アプリ、コントロールサービス層150および汎用OS121との関係を示すブロック図である。なお、図8では、アプリケーション130として、プリンタアプリ111、コピーアプリ112、新規アプリ117、118の例を示し、さらにこれら各アプリに対応したVAS 841, 842, 847および848を例として示しているが、他のアプリの場合も同様の構成である。

【0119】

また、実施の形態2にかかる複合機800では、実施の形態1の複合機100と異なり、図9に示すように、各VAS 841～848と各アプリとの間にはVAS制御プロセス（デーモン）801が動作している。

【0120】

このVAS制御プロセス（デーモン）801は、各アプリに対応したVAS 841～848を生成する。また、VAS制御プロセス801は、HD200に格納されたリソース使用情報ファイル201を参照して、起動要求が行われたアプリについて、リソース使用情報ファイル201にリソース使用情報が記録されているか否かをチェックして、リソース使用情報ファイルを作成する必要があるか否かを判断する。そして、起動要求が行われたアプリのリソース使用情報が記録されていない場合は、リソース使用情報取得スレッド141によって、アプリからの自己申告、あるいは、アプリを仮起動することにより、リソース使用情報を取得して、アプリ起動制限スレッド142に送るとともに、リソース使用情報フ

ファイル 201 を作成する。また、起動要求が行われたアプリのリソース使用情報が記録されている場合は、リソース使用情報ファイル 201 からリソース使用情報を読み出してアプリ起動制限スレッド 142 に送る。

【0121】

仮想アプリケーションサービス (VAS) 841~848 のプロセスには、ディスパッチャ 144 と、リソース使用情報取得スレッド 141 と、アプリ起動制限スレッド 142 とが動作している。

【0122】

ディスパッチャ 144 は、アプリケーション 130 やコントロールサービスからのメッセージ受信を監視し、受信したメッセージに応じてリソース使用情報取得スレッド 141、アプリ起動制限スレッド 142 に処理要求を行うものである。実施の形態 2 の複合機 800 では、ディスパッチャ 144 は、VAS 制御プロセス 801 から、アプリ ID、アプリのプロセス ID とともに、リソース使用情報取得要求メッセージまたはアプリ起動要求メッセージを受信するようになっている。ディスパッチャ 144 は、リソース使用情報取得要求メッセージを受信したときは、アプリ ID、アプリのプロセス ID とともに受信したリソース使用情報取得要求メッセージをリソース使用情報取得スレッド 141 に送信し、アプリ起動制限スレッド 142 からアプリ起動要求メッセージを受信したときは、アプリ ID、アプリのプロセス ID とともにアプリ起動要求メッセージをアプリ起動制限スレッド 142 に送信するようになっている。

【0123】

リソース使用情報取得スレッド 141 は、ディスパッチャ 144 からのリソース使用情報取得要求メッセージを受信すると、実施の形態 1 における VAS 140 の場合と同様に、起動アプリが使用するリソース使用量に関するリソース使用情報を取得して、リソース使用情報ファイル 201 をハードディスク (HD) 200 上に生成する。

【0124】

アプリ起動制限スレッド 142 は、ディスパッチャ 144 からのアプリ起動要求メッセージを受信すると、実施の形態 1 における VAS 140 と同様に、リソ

ース使用情報ファイル 2 0 1 を参照して、起動アプリが使用するリソース使用量と利用可能なリソース量とを取得して、比較することにより、少なくともリソース使用量がリソース量を越えた場合（越えなくても、リソース使用量がリソース量に近い場合）は、アプリ起動制限処理を行うようにする。

【 0 1 2 5 】

実施の形態 2 に複合機 8 0 0 における V A S 8 4 1 ～ 8 4 8 のリソース使用情報取得スレッド 1 4 1 によって実行されるリソース使用情報取得処理、およびアプリ起動制限スレッド 1 4 2 によって実行されるアプリ起動制限処理については、実施の形態 1 の複合機 1 0 0 における V A S 1 4 0 の各スレッドによる処理と同様である。

【 0 1 2 6 】

このように実施の形態 2 にかかる複合機 8 0 0 によれば、実施の形態 1 にかかる複合機 1 0 0 と同様に、複合機 8 0 0 のシステムの安定性を向上させることができる。

【 0 1 2 7 】

また、実施の形態 2 にかかる複合機 8 0 0 では、V A S 8 4 1 ～ 8 4 8 は起動されるアプリケーション 1 3 0 ごとに別個に起動されるので、複数のアプリケーション 1 3 0 に対する起動制限処理を各アプリケーション 1 3 0 に対応した V A S 8 4 1 ～ 8 4 8 によって並列に実行できることから、起動アプリケーションに対する起動制限処理を効率的に行うことができる。

【 0 1 2 8 】

なお、実施の形態 2 にかかる複合機 8 0 0 では、全てのアプリごとに別個に V A S 8 4 1 ～ 8 4 8 を起動していたが、一部のアプリに対してのみ V A S を起動するように構成しても良い。例えば、新規アプリ 1 1 7, 1 1 8 などサードベンダなどの第三者が開発したアプリに対してのみ V A S 8 4 7, 8 4 8 を起動してリソース使用情報の取得処理やアプリ起動制限処理を行い、プリンタアプリ 1 1 1 やコピーアプリ 1 1 2 などの既存のアプリに対してはこのようなサービスを行わないように構成することができる。

【 0 1 2 9 】

また、実施の形態 1 および 2 にかかる複合機 100, 800 では、リソースとして、テキストメモリ領域、ヒープ領域、スタック領域などを利用してリソース使用情報取得およびアプリ起動制限を行っていたが、かかるリソースは一例であって、これ以外のリソースを利用した構成としても勿論良い。

【0130】

【発明の効果】

以上説明したように、請求項 1 にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーションが起動される前に、アプリ起動制限手段によって、画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得し、取得したリソース量情報と、起動アプリケーションが使用するリソース使用量に関するリソース使用情報とに基づいて、リソース使用量がリソース量を越えている場合に起動アプリケーションの起動を制限するようにしたので、リソースの状況によって画像形成装置特有のアプリケーションの実行ができなくなることを未然に回避し、画像形成装置のシステムの安定性を向上させることができるという効果を奏する。

【0131】

また、請求項 2 にかかる発明によれば、リソース使用情報取得手段によって、画像形成装置に特有のアプリケーション以外の起動アプリケーションが使用するリソース使用量に関する情報を取得してリソース使用情報を生成することで、起動アプリケーションごとに実際に必要とするリソースに関するリソース使用情報に基づいてアプリ起動制限手段によるアプリ起動を制限するようにしたので、画像形成装置のシステムの安定性をより向上させることができるという効果を奏する。

【0132】

また、請求項 3 にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーションを起動する前に、リソース使用情報取得手段によって、各アプリケーションが自ら保有するリソース使用量に関する情報を取得し、リソース使用情報を生成するようにしたので、起動アプリケーションが使用する正確なリソース使用量を事前に把握することが可能となり、制限すべきアプリケー

ションの起動か否かを判断することができるという効果を奏する。

【0 1 3 3】

また、請求項 4 にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーションを通常起動する前に、リソース使用情報取得手段によって、起動前のアプリケーションを仮起動させることにより、そのアプリケーションが使用するリソース使用量に関する情報を取得することができ、ここからリソース使用情報を生成するようにしたので、起動アプリケーションが使用する正確なリソース使用量を事前に把握することが可能となり、制限すべきアプリケーションの起動か否かを判断することができるという効果を奏する。

【0 1 3 4】

また、請求項 5 にかかる発明によれば、リソース使用情報取得手段によって、起動アプリケーションが自ら保有するリソース使用量、あるいは、起動アプリケーションの仮起動によるリソース使用量に基づいて通常起動を行い、通常起動時に起動アプリケーションが実際に使用したリソース使用量に基づいてリソース使用情報を生成するようにしたので、起動アプリケーションが実際に使用するより正確なリソース使用量が把握できるようになり、当該アプリケーションが再度起動される前に、制限すべきアプリケーションの起動か否かを判断することができるという効果を奏する。特に、単独でのアプリケーションの起動では問題にならない程度のリソース使用量であっても、複数のアプリケーションを同時または追加的に起動する場合は、それらの合計リソース使用量が問題となるため、各アプリケーションが使用するリソース使用量をできるだけ正確に把握することが可能となる。

【0 1 3 5】

また、請求項 6 にかかる発明によれば、リソース使用情報として、通常起動時に起動アプリケーションが実際に使用したリソース使用量の平均値を求めるようにしたので、実体に則したリソース使用量が得られるようになり、アプリケーションの起動を必要以上に制限したり、逆に、リソース不足による画像形成装置の動作停止を防止することができるという効果を奏する。

【0 1 3 6】

また、請求項 7 にかかる発明によれば、リソース使用情報取得手段によって、起動アプリケーションが使用するメモリ領域情報を取得してリソース使用情報を生成し、アプリ起動制限手段によって、画像形成装置のメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するメモリ使用量を除いたメモリ領域残容量をリソース量情報として取得し、その取得したメモリ領域残容量とメモリ領域情報とに基づいて、起動アプリケーションの起動を制限するようにしたので、アプリケーションの実行に最も影響を及ぼすメモリ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができるという効果を奏する。

【0137】

また、請求項 8 にかかる発明によれば、リソース使用情報取得手段によって、起動アプリケーションが使用するテキストメモリ領域情報を取得してリソース使用情報を生成し、アプリ起動制限手段によって、画像形成装置におけるテキストメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するテキストメモリ使用量を除いたテキストメモリ領域残容量をリソース量情報として取得し、取得したテキストメモリ領域残容量とテキストメモリ領域情報とに基づいて、起動アプリケーションの起動を制限するようにしたので、アプリケーションのプログラムなどを格納するテキストメモリ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができるという効果を奏する。

【0138】

また、請求項 9 にかかる発明によれば、リソース使用情報取得手段によって、起動アプリケーションが使用するヒープ領域情報を取得してリソース使用情報を生成し、アプリ起動制限手段によって、画像形成装置におけるヒープ領域から画像形成装置特有のアプリケーションおよびシステムが使用するヒープ領域使用量を除いたヒープ領域残容量をリソース量情報として取得し、取得したヒープ領域残容量とヒープ領域情報とに基づいて、起動アプリケーションの起動を制限するようにしたので、アプリケーションが動的に確保または破棄するヒープ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避し

て、画像形成装置のシステムの安定性を向上させることができるという効果を奏する。

【0139】

また、請求項10にかかる発明によれば、リソース使用情報取得手段によって、起動アプリケーションが使用するスタック領域情報を取得してリソース使用情報を生成し、アプリ起動制限手段によって、画像形成装置におけるスタック領域から画像形成装置特有のアプリケーションおよびシステムが使用するスタック領域使用量を除いたスタック領域残容量をリソース量情報として取得し、取得したスタック領域残容量とスタック領域情報とに基づいて、起動アプリケーションの起動を制限するようにしたので、アプリケーションが関数呼び出し、サブルーチン呼び出しなどで使用するスタック領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができるという効果を奏する。

【0140】

また、請求項11にかかる発明によれば、リソース使用情報取得手段によって、プロセスが使用するリソースに関する情報を保持するシステム情報からアプリケーションが使用するリソースに関する情報を取得してリソース使用情報を生成するようにしたので、システム情報を参照するだけでアプリケーションが使用するリソースの使用情報が取得でき、リソース使用情報取得処理を効率的に行うことができる。

【0141】

また、請求項12にかかる発明によれば、アプリ起動制限手段によって、リソース量情報とリソース使用情報に基づいてアプリケーションに対し起動しないように制限を行う起動制限要求メッセージを送信するようにしたので、メッセージ送信というプロセス間通信を利用してアプリケーションの起動を制限することが可能となり、多数のプロセスが動作する画像形成装置においてアプリケーションの起動制限を効率的に行うことができるという効果を奏する。

【0142】

また、請求項13にかかる発明によれば、リソース使用情報取得手段と前記ア

プリ起動制限手段は仮想アプリケーションサービスに含まれており、仮想アプリケーションサービスがコントロールサービスをサーバとしたクライアントプロセスとして動作し、かつアプリケーションをクライアントとしたサーバプロセスとして動作するようにしたので、アプリケーションに対してリソース使用情報取得およびプリ起動制限というサービスを提供するだけでなく、コントロールサービスから画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得することができ、画像形成装置特有のアプリケーションの動作に影響を与えることなく、新規アプリケーションを搭載して利用することができるという効果を奏する。

【0143】

また、請求項14にかかる発明によれば、プリ起動制限手段とリソース使用情報取得手段を仮想アプリケーションサービスプロセス内のスレッドとして生成したので、複数のアプリケーションが並列に起動された場合に、リソース使用情報取得処理とプリ起動制限処理とをコンテキストの切り替えなしにCPUの占有切り替えによって並列実行することが可能となり、リソース使用情報取得処理とプリ起動制限処理との並列処理を円滑に行うことができるという効果を奏する。

【0144】

また、請求項15にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーションを起動する前に、アプリケーション毎に設けられた対応する仮想アプリケーションサービスのリソース使用情報取得手段とプリ起動制限手段とにより、起動するアプリケーションのリソース使用量と、利用可能なリソース量とを取得し、リソース使用量がリソース量を越えている場合にアプリケーションの起動を制限するようにしたので、リソース不足によりアプリケーションの実行ができなくなることを未然に回避し、画像形成装置のシステムの安定性を向上させることができるという効果を奏する。さらに、仮想アプリケーションサービスは、各アプリケーション毎に設けられ、アプリケーションごとに別個に処理可能なため、複数のアプリケーションの起動制限処理を各アプリケーションに対応する仮想アプリケーションサービスで並列に実行することにより、ア

アプリケーションの起動制限処理を効率的に行うことができるという効果を奏する。

【0145】

また、請求項16にかかる発明によれば、コントロールサービスは、ユーザサービスを提供する際に、アプリケーションの少なくとも2つが共通的に必要とするハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うようにしたので、複数のアプリケーションを作成する際に、共通化できる部分をコントロールサービス（プラットフォーム）として共通化し、それ以外の部分を作成すればよいことから、アプリケーションの開発が容易となり、かつ、短期間で開発することができるという効果を奏する。

【0146】

また、請求項17にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーションが起動される前に、アプリ起動制限ステップによって、画像形成装置特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得し、取得したリソース量情報と、起動アプリケーションが使用するリソース使用量に関するリソース使用情報とに基づいて、リソース使用量がリソース量を越えている場合に起動アプリケーションの起動を制限するようにしたので、リソースの状況によって画像形成装置特有のアプリケーションの実行ができなくなることを未然に回避し、画像形成装置のシステムの安定性を向上させることができるという効果を奏する。

【0147】

また、請求項18にかかる発明によれば、リソース使用情報取得ステップによって、画像形成装置に特有のアプリケーション以外の起動アプリケーションが使用するリソース使用量に関する情報を取得してリソース使用情報を生成するようにしたので、起動アプリケーションごとに実際に必要とするリソースに関するリソース使用情報に基づいてアプリ起動制限ステップによりアプリ起動を制限することが可能となり、画像形成装置のシステムの安定性をより向上させることができるという効果を奏する。

【0148】

また、請求項 19 にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーションを起動する前に、リソース使用情報取得ステップによって、各アプリケーションが自ら保有するリソース使用量に関する情報を取得し、リソース使用情報を生成するようにしたので、起動アプリケーションが使用する正確なリソース使用量を事前に把握することが可能となり、制限すべきアプリケーションの起動か否かを判断することができるという効果を奏する。

【0149】

また、請求項 20 にかかる発明によれば、画像形成装置に特有のアプリケーション以外のアプリケーションを通常起動する前に、リソース使用情報取得ステップによって、起動前のアプリケーションを仮起動させるようにしたので、そのアプリケーションが使用するリソース使用量に関する情報を取得することができ、ここからリソース使用情報を生成することにより、起動アプリケーションが使用する正確なリソース使用量を事前に把握することが可能となり、制限すべきアプリケーションの起動か否かを判断することができるという効果を奏する。

【0150】

また、請求項 21 にかかる発明によれば、リソース使用情報取得ステップによって、起動アプリケーションが自ら保有するリソース使用量、あるいは、起動アプリケーションの仮起動によるリソース使用量に基づいて通常起動を行い、通常起動時に起動アプリケーションが実際に使用したリソース使用量に基づいてリソース使用情報を生成するようにしたので、起動アプリケーションが実際に使用するより正確なリソース使用量が把握できるようになり、当該アプリケーションが再度起動される前に、制限すべきアプリケーションの起動か否かを判断することができるという効果を奏する。特に、単独でのアプリケーションの起動では問題にならない程度のリソース使用量であっても、複数のアプリケーションを同時または追加的に起動する場合は、それらの合計リソース使用量が問題となるため、各アプリケーションが使用するリソース使用量をできるだけ正確に把握することが可能となる。

【0151】

また、請求項 22 にかかる発明によれば、リソース使用情報として、通常起動

時に起動アプリケーションが実際に使用したリソース使用量の平均値を求めることにより、実体に則したリソース使用量が得られるようになり、アプリケーションの起動を必要以上に制限したり、逆に、リソース不足による画像形成装置の動作停止を防止することができるという効果を奏する。

【0 1 5 2】

また、請求項 2 3 にかかる発明によれば、リソース使用情報取得ステップによって、起動アプリケーションが使用するメモリ領域情報を取得してリソース使用情報を生成し、アプリ起動制限ステップによって、画像形成装置のメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するメモリ使用量を除いたメモリ領域残容量をリソース量情報として取得し、その取得したメモリ領域残容量とメモリ領域情報とに基づいて、起動アプリケーションの起動を制限するようにしたので、アプリケーションの実行に最も影響を及ぼすメモリ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができるという効果を奏する。

【0 1 5 3】

また、請求項 2 4 にかかる発明によれば、リソース使用情報取得ステップによって、起動アプリケーションが使用するテキストメモリ領域情報を取得してリソース使用情報を生成し、アプリ起動制限ステップによって、画像形成装置におけるテキストメモリ領域から画像形成装置特有のアプリケーションおよびシステムが使用するテキストメモリ使用量を除いたテキストメモリ領域残容量をリソース量情報として取得し、取得したテキストメモリ領域残容量とテキストメモリ領域情報とに基づいて、起動アプリケーションの起動を制限するようにしたので、アプリケーションのプログラムなどを格納するテキストメモリ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができるという効果を奏する。

【0 1 5 4】

また、請求項 2 5 にかかる発明によれば、リソース使用情報取得ステップによって、起動アプリケーションが使用するヒープ領域情報を取得してリソース使用

情報を生成し、アプリ起動制限ステップによって、画像形成装置におけるヒープ領域から画像形成装置特有のアプリケーションおよびシステムが使用するヒープ領域使用量を除いたヒープ領域残容量をリソース量情報として取得し、取得したヒープ領域残容量とヒープ領域情報とに基づいて、起動アプリケーションの起動を制限するようにしたので、アプリケーションが動的に確保または破棄するヒープ領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができるという効果を奏する。

【0155】

また、請求項 26 にかかる発明によれば、リソース使用情報取得ステップによって、起動アプリケーションが使用するスタック領域情報を取得してリソース使用情報を生成し、アプリ起動制限ステップによって、画像形成装置におけるスタック領域から画像形成装置特有のアプリケーションおよびシステムが使用するスタック領域使用量を除いたスタック領域残容量をリソース量情報として取得し、取得したスタック領域残容量とスタック領域情報とに基づいて、起動アプリケーションの起動を制限するようにしたので、アプリケーションが関数呼び出し、サブルーチン呼び出しなどで使用するスタック領域の容量不足などによってアプリケーションの実行ができなくなることを未然に回避して、画像形成装置のシステムの安定性を向上させることができるという効果を奏する。

【0156】

また、請求項 27 にかかる発明によれば、リソース使用情報取得ステップによって、プロセスが使用するリソースに関する情報を保持するシステム情報からアプリケーションが使用するリソースに関する情報を取得してリソース使用情報を生成するようにしたので、システム情報を参照するだけでアプリケーションが使用するリソースの使用情報が取得でき、リソース使用情報取得処理を効率的に行うことができるという効果を奏する。

【0157】

また、請求項 28 にかかる発明によれば、アプリ起動制限ステップによって、リソース量情報とリソース使用情報に基づいてアプリケーションに対し起動しな

いように制限を行う起動制限要求メッセージを送信するようにしたので、メッセージ送信というプロセス間通信を利用してアプリケーションの起動を制限することができ、多数のプロセスが動作する画像形成装置においてアプリケーションの起動制限を効率的に行うことができるという効果を奏する。

【図面の簡単な説明】

【図 1】

実施の形態 1 にかかる複合機の構成を示すブロック図である。

【図 2】

実施の形態 1 にかかる複合機の V A S の構成と、V A S と各アプリ、コントロールサービス層および汎用 O S との関係を示すブロック図である。

【図 3】

実施の形態 1 にかかる複合機におけるリソース使用情報ファイルの内容例を示す説明図である。

【図 4】

実施の形態 1 にかかる複合機における V A S のリソース使用情報取得スレッドが参照する p r o c 構造体の一例を示す説明図である。

【図 5】

複合機のオペレーションパネル上にユーザアプリの起動制限レベルをヒストグラムで表示した図である。

【図 6】

V A S の制御スレッドにより起動要求のあったアプリ起動制限処理手順を示すフローチャートである。

【図 7】

リソース使用情報ファイルの作成処理の動作を説明するフローチャートである。

【図 8】

実施の形態 2 にかかる複合機の構成を示すブロック図である。

【図 9】

実施の形態 2 にかかる複合機の V A S の構成と、V A S と各アプリ、コントロ

ールサービス層および汎用 OS との関係を示すブロック図である。

【符号の説明】

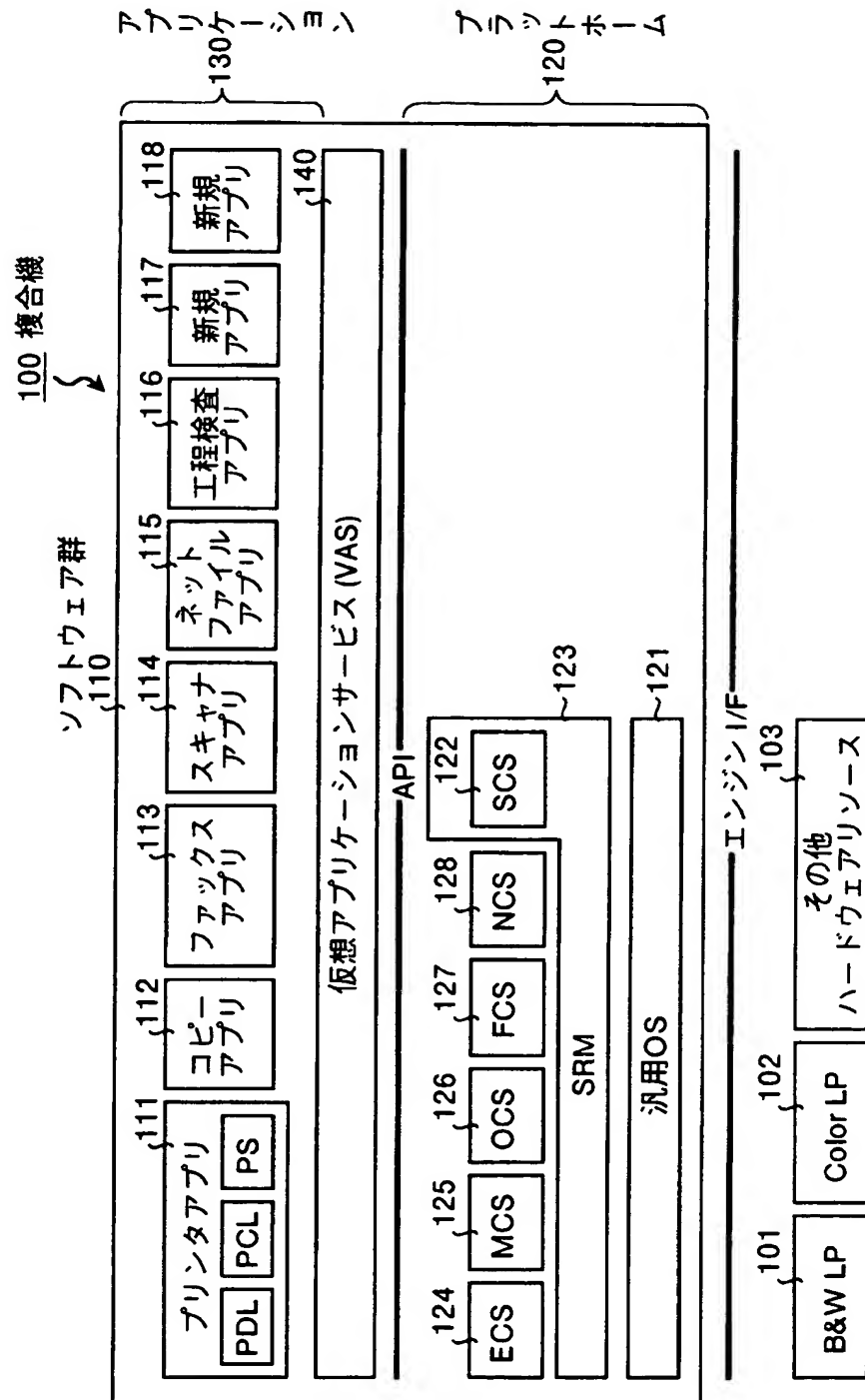
- 1 0 0 複合機
- 1 0 1 白黒レーザプリンタ
- 1 0 2 カラーレーザプリンタ
- 1 0 3 ハードウェアリソース
- 1 1 0 ソフトウェア群
- 1 1 1 プリンタアプリ
- 1 1 2 コピーアプリ
- 1 1 3 ファックスアプリ
- 1 1 4 スキャナアプリ
- 1 1 5 ネットファイルアプリ
- 1 1 6 工程検査アプリ
- 1 1 7, 1 1 8 新規アプリ
- 1 2 0 プラットホーム
- 1 2 1 汎用 OS
- 1 2 2 SCS
- 1 2 3 SRM
- 1 2 4 ECS
- 1 2 5 MCS
- 1 2 6 OCS
- 1 2 7 FCS
- 1 2 8 NCS
- 1 3 0 アプリケーション
- 1 4 0, 8 4 1 ~ 8 4 8 仮想アプリケーションサービス (VAS)
- 1 4 1 リソース使用情報取得スレッド
- 1 4 2 アプリ起動制限スレッド
- 1 4 3 制御スレッド
- 1 4 4 デイスパッチャ

- 1 5 0 コントロールサービス層
- 2 0 0 ハードディスク (H D)
- 2 0 1 リソース使用情報ファイル
- 2 1 0 R A M
- 2 1 1 p r o c 構造体
- 2 1 2 u 領域
- 2 2 1 ユーザアプリ
- 2 2 2 ユーザアプリ
- 2 2 3 ユーザアプリ
- 8 0 0 複合機
- 8 0 1 V A S 制御プロセス (デーモン)

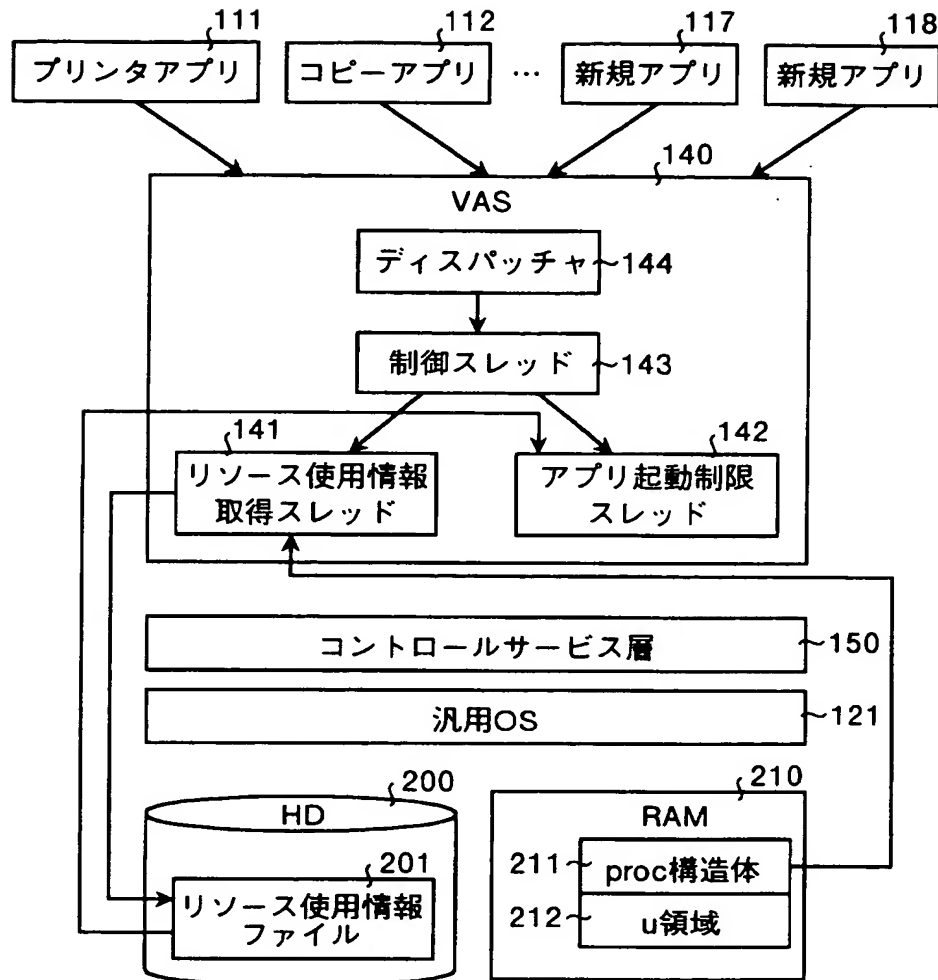
【書類名】

図面

【図 1】



【図 2】



【図 3】

リソース使用情報ファイル 201

アプリID	テキストメモリ 領域サイズ	ヒープ領域 サイズ	スタック領域 サイズ
101	3MB	1MB	10KB
102	2MB	500KB	5KB
103	4MB	200KB	3KB
・	・	・	・
・	・	・	・
・	・	・	・

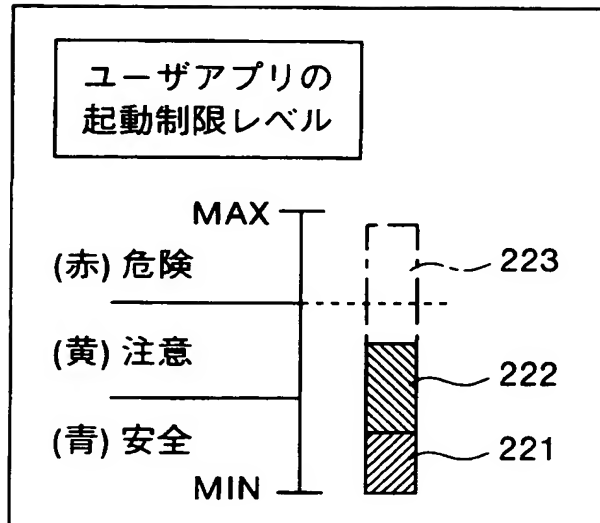
【図 4】

proc構造体 211

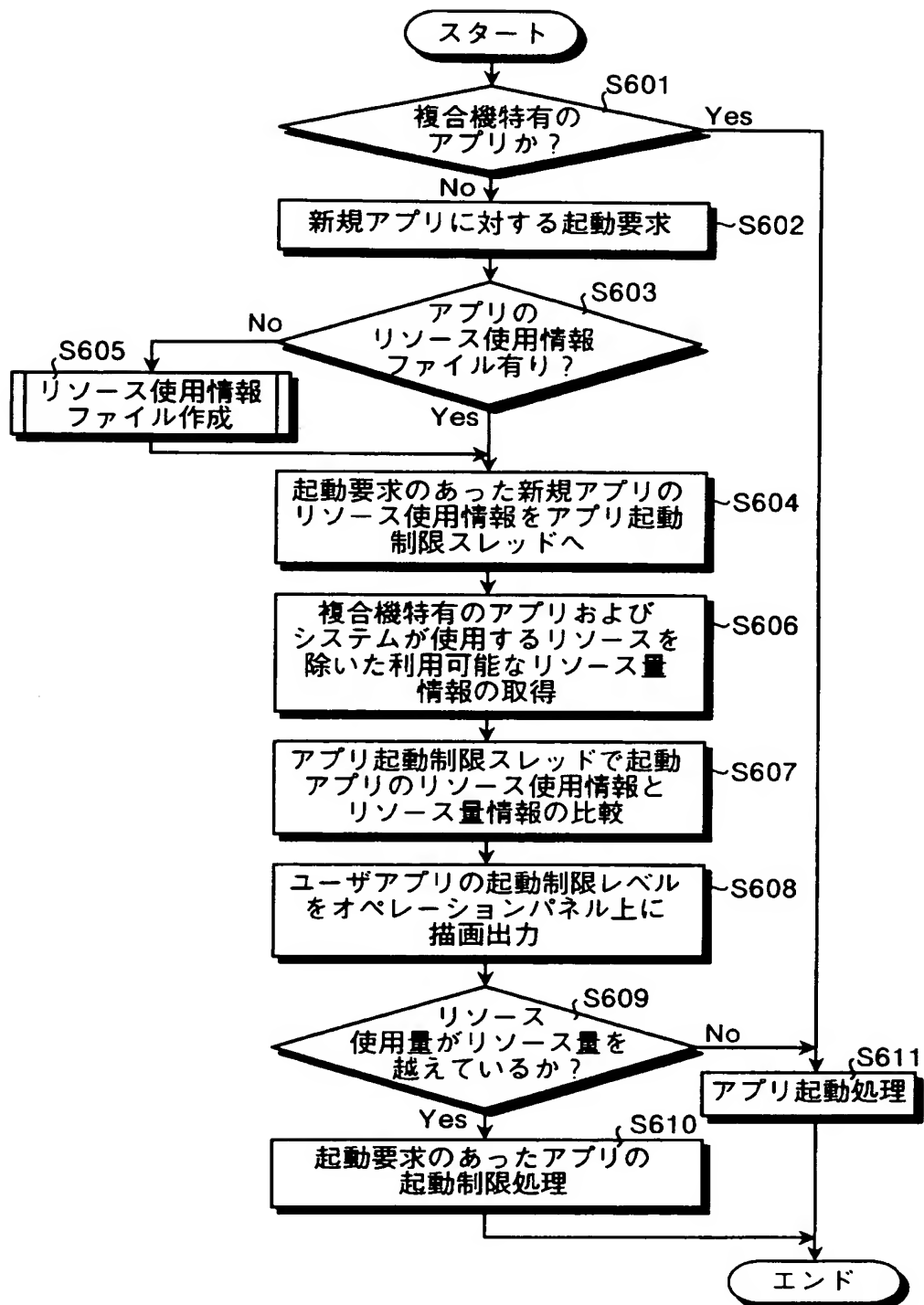
メンバ名	説明
p_pid	プロセスID
p_txtsz	テキストメモリ領域サイズ
p_heapsz	ヒープ領域サイズ
p_stacksz	スタック領域サイズ
・	・
・	・
・	・

【図 5】

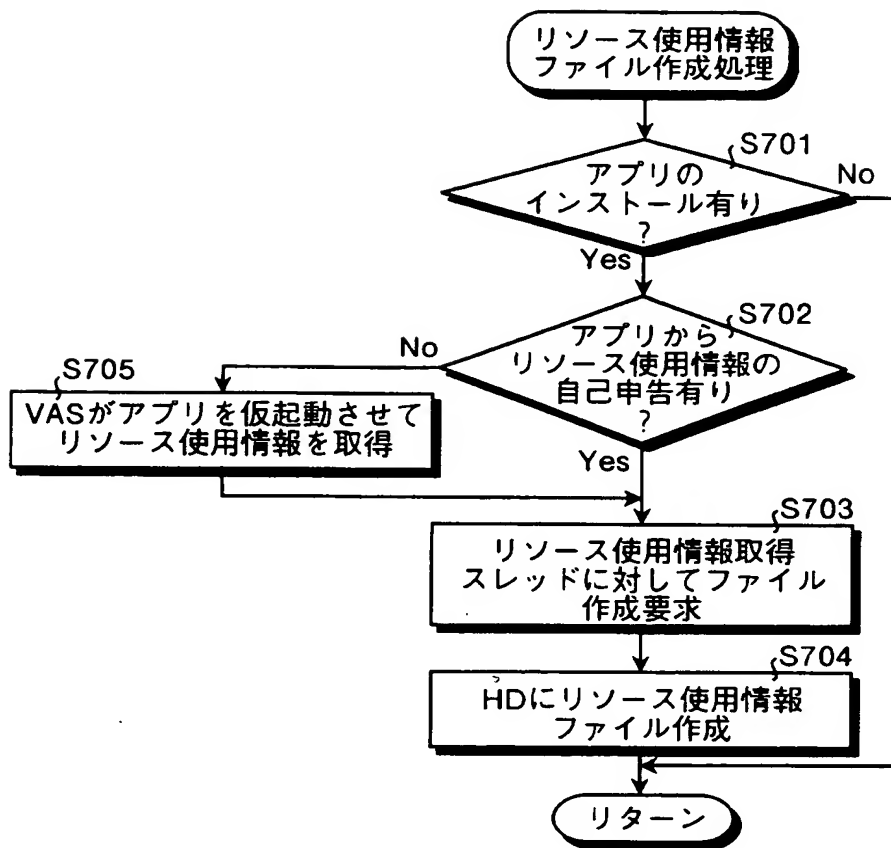
オペレーションパネル (ヒストグラム)



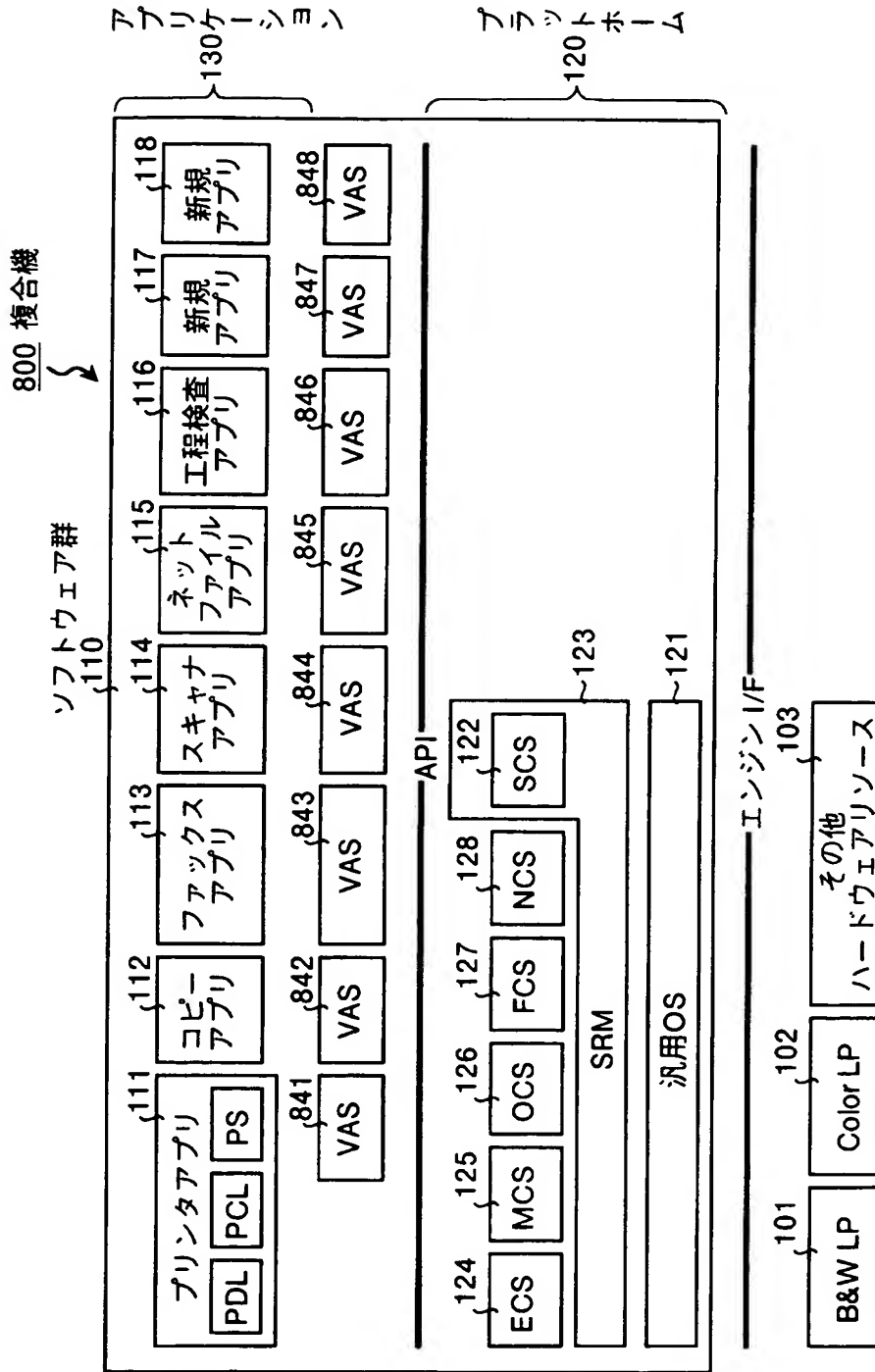
【図 6】



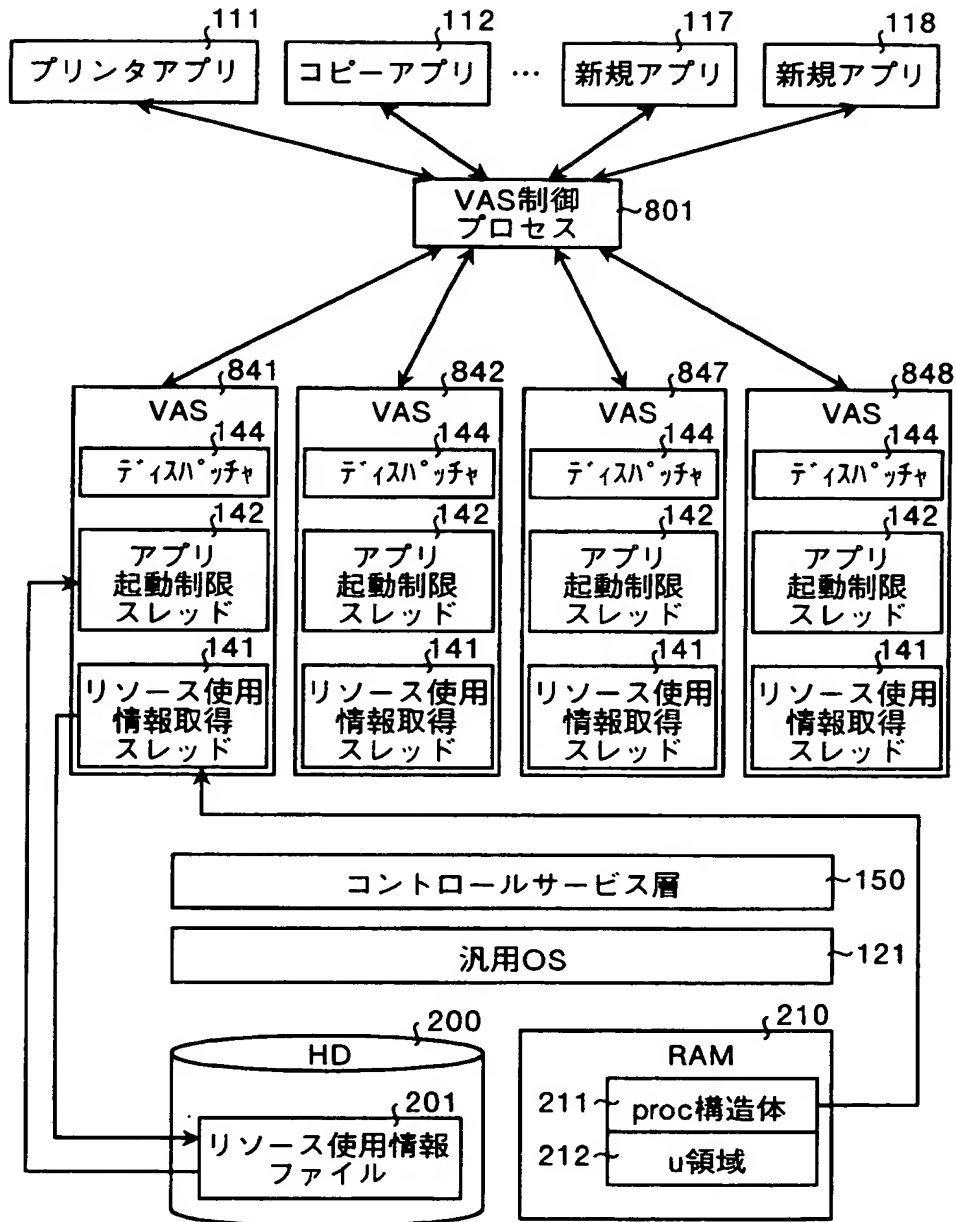
【図 7】



【図 8】



【図 9】



【書類名】 要約書

【要約】

【課題】 複合機特有のアプリケーションを常に安定した状態で確実に動作させることができるようにする。

【解決手段】 複合機に特有のアプリケーション以外のアプリケーションを起動する前に、起動アプリケーションが使用するリソース使用量に関するリソース使用情報を取得するリソース使用情報取得スレッド141と、複合機特有のアプリケーションおよびシステムが使用するリソースを除いた利用可能なリソース量情報を取得し、そのリソース量とリソース使用量とを比較して、リソース使用量がリソース量を越えている場合は起動アプリケーションの起動を制限するアプリ起動制限スレッド142とを有する仮想アプリケーションサービス（VAS）140を備えている。

【選択図】 図2

特願 2 0 0 2 - 2 6 9 2 8 0

出 願 人 履 歷 情 報

識別番号

[0 0 0 0 0 6 7 4 7]

1. 変更年月日

1 9 9 0 年 8 月 2 4 日

[変更理由]

新規登録

住 所

東京都大田区中馬込 1 丁目 3 番 6 号

氏 名

株式会社リコー

2. 変更年月日

2 0 0 2 年 5 月 1 7 日

[変更理由]

住所変更

住 所

東京都大田区中馬込 1 丁目 3 番 6 号

氏 名

株式会社リコー